SYSTEMS ANALYSIS OF METABOLISM AND PHYSIOLOGY IN THE OIL-PRODUCING GREEN ALGA *BOTRYOCOCCUS BRAUNII* RACE B (SHOWA)

A Dissertation

by

DANIEL R BROWNE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Timothy P. Devarenne
Committee Members,	Rodolfo A. Aramayo
	Paul A. Lindahl
	John E. Mullet
Head of Department,	Dorothy E. Shippen

December 2018

Major Subject: Biochemistry

ABSTRACT

The colony-forming green microalga *Botryococcus braunii* is mostly known for its ability to produce an abundance of liquid hydrocarbons. However, geochemical studies have found fossilized remains of the species in petroleum source rocks from around the planet, dating as far back as the Precambrian eon. Thus *B. braunii* is considered a source of petroleum throughout the geological ages and presents an interesting model to study hydrocarbon metabolism.

To better understand the biochemical and genetic systems that underpin the unique properties of *B. braunii*, we have sequenced and analyzed its genome. Using a comparative genomics approach, we identified 187 functions that are unique in *B. braunii* among the Viridiplantae (green algae and land plants), and 402 functions that are unique in *B. braunii* among the green algae. Thus *B. braunii* shares 215 functions with land plants that other green algae do not. These functions include parts of the photosynthetic apparatus, the ubiquitin system, cytochrome P450s, peptidases, cytoskeleton proteins, and others.

To further understand the active interpretation of genomic information, we sequenced the transcriptome of *B. braunii* every six hours over the course of three days. The goal of this experiment was to determine the gene expression patterns associated with light/dark transitions. Interestingly, we found several strong coexpression modules that cycle, not according to light or dark conditions, but by time of day, indicating the presence of circadian regulatory mechanisms.

To determine the impact of gene expression and time of day on metabolism, we generated metabolomics data for each of the biological samples that were utilized to obtain the transcriptome data. Targeted and untargeted analyses of polar and nonpolar metabolites revealed that unlike transcription, metabolite pools do not appear to significantly change with time of day.

The information presented in this dissertation adds great value to our fundamental understanding of the systems governing *B. braunii* metabolism and physiology. With this knowledge, we could design genetic systems in heterologous hosts to mimic the properties of *B. braunii* pathways. This could result in synthetic pathways for hydrocarbon production with strong metabolic flux, technology that is vital for the development of sustainable bioproducts.

DEDICATION

This dissertation is dedicated to my family, especially my mother Susan Forter Browne, my father Norman Browne, my aunt Jody Forter, and my grandmother Joan Forter. Your love and support have made all of this possible. Thank you for believing in me throughout my life.

ACKNOWLEDGEMENTS

Thanks first and foremost to my advisor, Dr. Timothy Devarenne, for investing in me and giving me the opportunity to grow into the scientist that I am today. He gave me the freedom to explore that enabled me to begin learning about bioinformatics and he supported me throughout the learning process. I appreciate his trust in my abilities and his patience while I built up the skills required to achieve the work presented in this dissertation.

I am also grateful to each of my committee members, who have all been supportive in different ways. Dr. John Mullet allowed me to work in his lab in the summer months of 2012, at the beginning of my graduate school journey, giving me my first taste of doctoral life and graduate level research. Dr. Paul Lindahl and I share an interest in the origin of life and I have always enjoyed our discussions on this topic, as well as his perspective on my work. Dr. Rodolfo Aramayo has been especially supportive, on a personal as well as intellectual level, and I am deeply grateful for his guidance, encouragement, and friendship over the years - he introduced me to genomics and I would not be where I am today without him.

To the members of the Devarenne Lab I extend my thanks. Dr. Mehmet Tatli was my closest friend throughout this journey and words cannot express how grateful I am that our paths in life crossed. Dr. Hem Thapa was also a close confidant, and we shared many great scientific and philosophical discussions, working side by side in the laboratory. Dr. Dongyin Su and Incheol Yeo contributed valuable perspectives in lab meetings over the years.

There are many others who I would like to thank, including my family, my friends here in College Station, in Portland, and in San Diego, the BGA, the NGSB, and others. Thanks also to the many kind scientists with whom I shared valuable discussions over the Internet.

CONTIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Professors Dr. Timothy P. Devarenne [advisor] of the Department of Biochemistry and Biophysics, Dr. John E. Mullet of the Department of Biochemistry and Biophysics, Dr. Paul A. Lindahl of the Department of Chemistry and the Department of Biochemistry and Biophysics, and Dr. Rodolfo Aramayo of the Department of Biology.

Most of the work for this dissertation was completed by the student, under the supervision of Dr. Timothy P. Devarenne of the Department of Biochemistry and Biophysics. The genome libraries described in section 2 were prepared and sequenced by Dr. Jennifer Chiniquy and Dr. Aditi Sharma at the Joint Genome Institute and by Dr. Jane Grimwood at the HudsonAlpha Institute for Biotechnology. The version 0.5 and 1.0 genome assemblies described in section 2 were done by Dr. Jerry Jenkins and Jeremy Schmutz of the HudsonAlpha Institute for Biotechnology. The predicted *B. braunii* genome annotations described in sections 2, 3 and 4 were done by Dr. Shengqiang Shu and Dr. David Goodstein of the Joint Genome Institute. The transcriptome libraries described in section 4 were prepared and sequenced by Dr. Yuko Yoshinaga at the Joint Genome Institute. Project management functions for the DNA and RNA sequencing projects described in sections 2 and 4 were carried out by Kerrie Barry at the Joint Genome Institute. The metabolomics data described in section 4 were generated by Dr. Katherine Louie and Dr. Benjamin P. Bowen at the Joint Genome Institute. Project management functions for the metabolomics project described in section 4 were carried out by Dr. Katherine Louie and Dr. Trent Northen at the Joint Genome Institute.

Funding Sources

This work was made possible in part by NSF-EFRI-PSBR under grant #1240478 to Dr. Timothy P. Devarenne; by the JGI 2010 Community Sequencing Program, grant CSP2010-784140 to Andrew T. Koppisch, Timothy P. Devarenne, Joe Chappell, and David T. Fox; and by NSF grant CHE-1412648 to Andrew T. Koppisch. The work conducted by the U.S. Department of Energy Joint Genome Institute was supported by the Office of Science of the U.S. Department of Energy under contract no. DE-AC02-05CH11231.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

The following sections will provide broad overviews of the evolution of Viridiplantae, the history of *B. braunii* research, and recent advances in algae cultivation and sustainable biotechnology. The goal is to inform the reader of the bigger pictures and give essential context for much of the discussion in the subsequent sections.

1.1 Origins and Evolution of Viridiplantae

This section tells the comprehensive story of Viridiplantae evolution, from the origin of life to the current panoply of plants and algae that inhabit Earth. It is important to consider the complete line of evolution that connects all of the Viridiplantae, along with all other living organisms. This perspective will enable a better understanding of the results and discussion of the comparative genomics analyses presented in Section 3.

1.1.1 Prebiotic Chemistry and the Transition to Life

The beginning of life is one of the greatest and most captivating scientific mysteries. Enshrouded in the darkness of deep time, there are only tiny fragments of evidence that remain. The earliest chemical evidence of life on Earth was found in Eoarchean rocks, dated to approximately 3.95 Ga (gigayears ago), sourced from Labrador, Canada (*1*). Analysis of carbon isotope ratios in graphite isolated from these sediments indicates a potential biogenic origin. In particular, the isotopic fractionation between graphite and carbonate suggests the existence of autotrophic metabolism through a mechanism resembling either the reductive acetyl-CoA pathway or the Calvin cycle. The earliest fossil evidence of life on Earth was found in ferruginous (iron-rich) sedimentary rocks from Quebec, Canada, which originated as seafloor-hydrothermal vents

approximately 3.77-4.28 Ga (*2*). These putative microfossils have similar morphology and mineral contents to those of contemporary filamentous microorganisms and confirmed microfossils from younger rocks. The mineralogical components of the sediments suggest the existence of oxidation-reduction reactions, centered on the conversion of carbonaceous material and ferric iron into ferric-ferrous silicates, apatite, magnetite, and carbonate. While the oceans of the Precambrian Eons were ferruginous, such water bodies are rarely found on Earth today. However, one ferruginous system, Kabuno Bay in the Democratic Republic of Congo, provides an excellent model that is analogous to the ancestral ferruginous oceans. Kabuno Bay hosts a rich community of pelagic photoferrotrophs, which capture sunlight to oxidize ferrous iron and fix inorganic carbon into biomass, in turn supporting an ecosystem of heterotrophic microbes (*3*). Today, iron cycling in the ocean is tightly coupled to the availability of the major nutrients (nitrogen, carbon, phosphorous), and plays a critical role in regulating primary productivity (*4*). Evidence of ancient life has also been found in sedimentary remnants of mild environments, such as the 3.43 Ga Strelley Pool Formation in Western Australia, which originates from the oldest known shoreline (*5*). Novel analytical techniques based on ion beam milling and transmission electron microscopy have enabled the 3D reconstruction and identification of hitherto unknown microfossils, revealing some of the earliest known cyanobacteria-like cellular structures. Isotopic and mineralogical evidence associated with these structures indicate the possible existence of anoxygenic photosynthetic pathways involving the oxidation of hydrogen sulfide.

With such sparse availability of evidence, knowledge of early life is highly dependent on the development of theoretical mechanisms. For example, it is hypothesized that a pre-cellular, chemoautotrophic organism would consist of an inorganic substructure and an organic superstructure (*6*). The inorganic component could theoretically arise from volcanic materials,

including transition metals such as iron, cobalt, and nickel, providing surfaces and catalytic activities for the organic components. Through the development of an autocatalytic feedback mechanism, evolution over many generations could lead to the emergence of cellularization and heritable genetic information. This hypothesis is formally known as the Iron-Sulfur World theory and is the strongest and most comprehensive argument for a chemoautotrophic origin of life (*7*). This metabolism-first approach demonstrates plausible biochemical pathways that could emerge from simple chemical phenomena occurring on mineral surfaces. It is built on top of the theory of surface metabolism, which provides a logical foundation for the development of prebiotic biochemical pathways and the emergence of autocatalytic metabolic cycles (*8*). The other major theory for the origin of life is the RNA World theory (*9*). This article thoroughly reviews the potential pathways from RNA to DNA and protein, and also considers the possibility that DNA preceded RNA. While it is generally accepted that RNA was once the primary catalytic and informational molecule of life, there is still much debate on what preceded the RNA World and how life transitioned to the current DNA/RNA/protein system. The key question about the RNA world is how it became established. This question is addressed in a review that focuses on the origin and evolution of the RNA World prior to the development of protein synthesis (*10*). It provides a rich discussion of prebiotic chemistry and the potential synthetic mechanisms of early nucleic acids. Interestingly, nucleobases were discovered in formic acid extracts of 12 different meteorites (*11*). Furthermore, the authors conducted synthetic experiments with ammonium cyanide and revealed the formation of identical nucleobases as found in the meteorites. This evidence demonstrates a chemical mechanism for the prebiotic formation of nucleobases in asteroid bodies and supports the possibility of a chemically spontaneous RNA World.

Chemical experimentation is a critical aspect of developing and testing hypotheses about the origin of life. In a classical experiment, Miller and Urey found that a mixture of methane, ammonia, water, and hydrogen, modeled on the early atmosphere, gives rise to the formation of simple amino acids in the presence of an electrical discharge (*12*). This simple and powerful experiment demonstrates a potential prebiotic mechanism for the synthesis of molecules that are essential for the development of life. Since Miller's original experiments in 1953, the development of incredibly powerful analytical tools such as liquid chromatography-mass spectrometry has transformed the ability to detect minute amounts of molecules. Just before his death in 2007, some of his original experimental samples were found in his laboratory and analyzed with modern equipment (*13*). This led to the discovery of more than 40 different amino acids and amines, which had never before been detected. Given the abundance of hydrogen, methane, ammonia, and water in the Universe, Miller and Urey showed it is possible that the molecules of life could spontaneously accumulate in many places. However, while they assumed that the early atmosphere contained hydrogen, methane, ammonia, and water, this is a photochemically labile mixture and unlikely to have persisted in the early atmosphere of Earth (*14*). Studies based on solid Earth outgassing argued that the original atmosphere was composed mostly of water, carbon dioxide, and nitrogen; with only a little carbon monoxide and hydrogen, and virtually no methane or ammonia. Another approach suggested that the early atmosphere accumulated by gas emissions from impacting material, which could include molecules such as carbon monoxide, cyanide, ethylene, ethane, sulfur, and others. It is important to consider that hydrogen, water, methane, and ammonia are commonly found in the Universe, often constitute planetary atmospheres, like those of Jupiter, Saturn, Uranus, and Neptune, and could have been major components of Earth's early

atmosphere. There is essentially no evidence left of the early atmosphere and so we can only make estimates based on theoretical developments and observation of extraterrestrial systems.

The work of Miller and Urey formed the basis of the RNA World theory and Miller spent much of the rest of his career expounding on this theory. Yet recent chemical experiments have provided very interesting insights into the plausibility of the Iron-Sulfur World theory. The Krebs cycle is one of the most highly conserved and central enzyme-catalyzed metabolic pathways in extant life. Recently, it was discovered that sulfate radicals from peroxydisulfate enabled non-enzymatic catalysis of Krebs cycle intermediates, especially in the presence of ferrous sulfide (*15*). This evidence is incredibly important for the Iron-Sulfur World theory of surface metabolists. However, the key question in the origin of life appears to be centered around the early primacy of genetic machinery or metabolic pathways. One hypothesis is that the evolution of metabolism gave rise to the emergence of genetic processes (*16*). With the encapsulation of metabolic networks in cellular systems, genetic evolution began to drive the course of life. In contrast to the arguments for the sequential emergence of either metabolic (Iron-Sulfur World) or genetic (RNA World) systems, recent work suggests simultaneous emergence. Elegant chemical experiments demonstrated the synthesis of precursors of ribonucleotides, amino acids, and lipids from the same starting materials of hydrogen sulfide, hydrogen cyanide, and derivates thereof (*17*). This points to the coevolution of prebiotic chemicals into complex systems of coupled components, leading to the emergence of life. Another step in the direction of unifying the RNA World and Iron-Sulfur World hypotheses is the concept of metabolically coupled replicator systems (MCRSs). This model details the development of autocatalytic cycles that self-replicate, propagate, and evolve over time, meeting the basic criteria for life (*18*). Despite this progress, the emergence of membrane-bound cells is one of the major questions remaining in the origin of life. Researchers

were able to gain some insights into ancestral membrane evolution by examing the membranes of archaea and bacteria, and suggest that the last universal common ancestor may have had membranes with the characteristics of both groups (*19*). Although the Iron-Sulfur World and RNA World hypotheses provide chemical and genetic mechanisms for prebiotic systems, respectively, it is unclear how these mechanisms could transition to living systems. One scientist attempted to solve this problem by proposing theoretical steps by which a nonliving system could become a living system through combinatorial and selective processes (*20*).

While the first life forms were prokaryotic archaeal and bacterial single-celled organisms, the emergence of eukaryotes transformed the course of evolution and enabled the development of greater complexity (*21*). It is hypothesized that eukaryotes emerged from an archaeal host and an alphaproteobacterial endosymbiont (i.e. mitochondria). Archaeal and bacterial genomes are critical for obtaining clues as to the origins of contemporary cellular complexity. However, prior to the establishment of heritable genomes, the emergence of genes is another important and open question in the study of abiogenesis. Recent work has demonstrated that certain random polymers of DNA can confer fitness benefits when expressed in bacteria (*22*). This area of study can help to build understanding of how nucleic acid polymers were initially selected and propagated. A major difficulty in obtaining knowledge of this process is that the selective pressures on the genetic code of ancient organisms are very different from those of modern organisms (*23*). Evolution of the genetic code is essentially separated into two phases: before and after the establishment of the standard genetic code. Thus our knowledge of contemporary genetic evolution may not be at all informative in the investigation of ancient genetic evolution. Even after the establishment of the standard genetic code, there are still substantial challenges in resolving the course of evolution and taxonomically classifying organisms. The ability of organisms to exchange genetic material, such

as through horizontal gene transfer, complicates the effort to construct taxonomic hierarchies (*24*). New approaches are integrating whole-genome information to identify taxon-specific genomic signatures and clarify phylogenetic relationships. Such analyses could be greatly aided by the addition of information from ancient genomes, with new techniques in the extraction and sequencing of ancient DNA enabling revolutionary insights into genomic evolution (*25*). However, this approach is still quite limited in terms of geological time, only allowing researchers to reach back in time perhaps as far as one million years. Nonetheless, this information would be particularly useful for improving understanding of the mutation rate of genetic sequences, a critical force in evolution. The mutation rate of DNA varies widely across species and genes and it has been suggested that there are evolutionary constraints on the mutation rate (*26*). Quantitative modeling supports studies of the mutation rate and its impact on evolution. Moreover, mathematical models of gene and genome evolution are important for developing our understanding of the mechanisms of genetic evolution. Recently, a new model was developed describing genome evolution by transformation, expansion and contraction, based on substitution, insertion, and deletion of genetic motifs (*27*). This model is significant in that it includes mutations beyond simple single nucleotide polymorphisms. However, models of evolution must also move beyond the raw genetic sequence and consider the proteins and enzymes encoded by the genomes. Studying the evolution of enzymes requires a robust system of classifying structure and function (*28*). Enzymes are grouped into families and superfamilies based on primary sequence. The chemistry of enzymes is described by the Enzyme Commission (EC) system developed in 1956 by the International Union of Biochemistry and Molecular Biology (IUBMB). The accumulation of huge amounts of data over the years has revealed that many enzymes catalyze multiple reactions and that enzymes within one superfamily can catalyze much different chemistry.

Ultimately, the basis of life is the energy input it takes to sustain itself. Recently, it was argued that there are five major energy epochs in the history of Earth that underpin and explain the evolutionary expansions of the biosphere (*29*). Thus, a holistic understanding of the origin and evolution of life requires the integration of energy sources, chemical mechanisms, genetic mechanisms, and selective pressures. Although much progress has been made, further work is needed to combine competing and complementary theories, to test the thermodynamic and kinetic validity of the underpinning assumptions, and demonstrate chemical and genetic mechanisms. This will help to answer the most fundamental questions about the origin of life, yield essential information about the basal functions required for life, and provide important context for the subsequent evolution of complex biota.

1.1.2 Cyanobacteria and Evolution of Chloroplasts

The emergence of oxygenic photosynthesis is one of the most important events in the evolution of life on Earth. The structure and function of photosystems I and II (PSI and PSII), the principle reaction centers of oxygenic photosynthesis, have been thoroughly reviewed (*30*). Briefly, PSII utilizes light energy to oxidize water, extracting electrons and releasing hydrogen ions and molecular oxygen. The electrons are received by plastoquinone, carried to the cytochrome b6f complex, and transferred to plastocyanin. PSI utilizes light energy to oxidize plastocyanin, driving electron transfer to ferredoxin, which is subsequently used to generate NADPH by ferredoxin-NADP+ reductase. Simultaneously, the action of PSII and the cytochrome b6f complex build a gradient of hydrogen ions that drives the synthesis of ATP. The NADPH and ATP molecules produced by photosynthesis provide the energy required for fixing carbon dioxide and supporting metabolism, while the molecular oxygen diffuses out of the cell.

Approximately 2.45 Ga the atmosphere radically changed in what is known as the Great Oxidation Event (GOE), with the substantial accumulation of atmospheric oxygen, almost certainly due to the evolution of oxygenic, photosynthetic cyanobacteria (*31*). Some of the earliest known fossil evidence of cyanobacteria, occurring in Archean tufted microbial mats, has been dated at 2.72 Ga, well before the GOE. The apparent gap of approximately 200-300 million years between supposed the evolution of cyanobacteria and the GOE presents an important question: why did it take so long for molecular oxygen to accumulate in the atmosphere? One recent modeling effort estimated that it would only take approximately 100,000 years to populate the atmosphere with oxygen after the emergence of cyanobacteria (*32*). These results suggest that oxygenic photosynthesis evolved shortly before the GOE and not several hundred million years prior. Separately, it was estimated that the evolution of cyanobacteria from the prebiotic soup could have occurred within in a timespan of no more than 10 million years (*33*). The authors argue this is significant because it is possible that early life was wiped out approximately 3.8 Ga by the impacts of large asteroids. Traditionally, it was assumed that the transition from prebiotic soup to cyanobacteria took a very long time, on the order of a billion years or so. However, the estimates of atmospheric oxygenation and cyanobacterial evolution again beg the question of why it took so long for the atmosphere to actually accumulate significant amounts of oxygen. Interestingly, analysis of ancient micrometeorites suggested that the upper atmosphere of the Archean era was in fact oxygen-rich, in contrast to the oxygen-poor lower atmosphere (*34*). It was suggested that this observation indicates minimal mixing between the upper and lower atmospheres during the Archean. Analysis of chromium isotopes from ancient banded iron formations reveals redox conditions consistent with the timeframe of the GOE (*35*). However, this analysis also found that there appears to be a decrease in atmospheric oxygen approximately 1.88 Ga, meaning the GOE

did not lead to a unidirectional and stepwise accumulation of oxygen. One interpretation of this data is that there were fluctuations in the population size of cyanobacteria over the early eons that in turn led to variable levels of atmospheric oxygen.

Genomics offers insights into the evolution of cyanobacteria that can help improve our understanding of early atmospheric oxygenation. Comparative analysis of 15 cyanobacterial genomes yielded insights into the origin and evolution of oxygenic photosynthesis (*36*). It was concluded that anoxygenic phototrophic bacteria were the direct precursors of cyanobacteria and that oxygenic photosynthesis evolved under the selective pressures of ultraviolet light and depletion of electron donors. Furthermore, the acquisition of new genes by the processes of duplication and divergence enabled the cyanobacteria to oxidize water, releasing molecular oxygen. Another comparative analysis of cyanobacterial genomes reveals that horizontal gene transfer has affected approximately 60% of their genes (*37*). Additionally, the authors conclude that the phylogenetic evidence supports the hypothesis that oxygenic photosynthesis evolved in a freshwater environment. This conclusion is especially interesting, since freshwater is a much smaller environmental niche than the oceanic environment; it could help to explain why it took such a long time to oxygenate the atmosphere. That is, if oxygenic cyanobacteria were limited to freshwater environments, they would not be able to accumulate much biomass, relative to the oceanic environment. If it took a long time for oxygenic cyanobacteria to adapt to the oceanic environment and spread across the globe, this could account for the aforementioned gap between the first observed cyanobacteria and the oxygenated atmosphere. To better understand the core ancestral cyanobacterial genome, essential genes were identified by the creation and sequencing of a transposon mutant library (*38*). The authors identified as essential 718 protein-coding genes, 13 non-coding genes, 138 regulatory regions, and 15 other intergenic regions. Surprisingly, they

found that certain genes in the TCA cycle were non-essential. Furthermore, only a subset of the photosynthetic genes were classified as essential, likely representing the enzymes that comprise the ancestral photosynthetic machinery that was inherent from non-oxygenic photoautotrophic progenitors of cyanobacteria. In a separate study, it was revealed that photorespiratory 2-phosphoglycolate (2PG) metabolism is essential in cyanobacteria, as it is in plants (*39*). This suggests that 2PG metabolism is an essential partner for oxygenic photosynthesis. To better understand cyanobacterial population dynamics, a sample of wild *Prochlorococcus* was collected and analyzed with single-cell genomics tools, revealing immense genetic diversity (*40*). The authors found that the community was partitioned into distinct subpopulations defined by their associated "genomic backbones" consisting of core gene alleles linked to smaller sets of flexible genes. The results indicated that *Prochlorococcus* evolution is governed primarily by selection and not by genetic drift.

The development of photosynthetic eukaryotes is the next major step in the evolutionary progression towards modern plants. It is believed that photosynthetic eukaryotes are derived from a single primary endosymbiotic event where an ancestral heterotrophic eukaryote engulfed an ancestral cyanobacterium, forming the base of the Archaeplastida (*41*). The species in this group are characterized by a plastid surrounded by exactly two membranes. Phylogenomic analysis of plastid- and nucleus-encoded genes of cyanobacterial ancestry provides evidence for a deep origin of plastids within the cyanobacteria (*42*). Significantly, the results of this study support the hypothesis that photosynthetic eukaryotes emerged in a freshwater environment and not in the ocean. Subsequently, this basal photosynthetic eukaryote differentiated into the Glaucophyta, Rhodophyta, and Viridiplantae. Genomic analysis of *Cyanophora paradoxa*, one of the glaucophytes that emerged after the primary endosymbiosis, provides evidence in support of a

single primary endosymbiotic event, occurring more than 1 Ga (*43*). While the current view on the primary endosymbiotic event that led to plastids is that it occurred only once, approximately 1.5 Ga, it is also well established that there have been numerous secondary and tertiary endosymbiosis events among eukaryotes (*44*). Species arising from post-primary endosymbiosis are demarcated by a plastid surrounded by three or more membranes. The consequences of this include the potential for nucleus-to-nucleus gene transfer between hosts and endosymbionts, further complicating the effort to resolve evolutionary relationships among plastid-bearing eukaryotes. The evidence for the widespread occurrence of horizontal gene transfer throughout the history of evolution has been thoroughly reviewed (*45*). Genes have been exchanged not only within but also across the domains of life, between Archaea, Bacteria, and Eukaryotes. This phenomenon makes it incredibly difficult to construct a universal tree of life that depicts the course of evolution.

The evolution of plastids is a complex process and our understanding of it is confounded by the occurrence of horizontal gene transfer as well as endosymbiotic gene transfer. In a recent review, the authors discussed plastid distribution in eukaryotes, summarized the diversity of photosynthetic pigments and carbon storage mechanisms, and discussed the latest advances in our holistic understanding of plastid evolution (*46*). The exchange of genetic material between plastids and their host cells has led to a tight functional integration between the two, as recently reviewed (*47*). Briefly, metabolic redundancy between plastids and the host cells has been eliminated by consolidation. Endosymbiotic gene transfer has led to the host nuclear and cytoplasmic production of many plastid-originating proteins that are subsequently targeted back to the plastid. Furthermore, some plastid-originating proteins produced by the nucleus are no longer targeted to the plastid, but are delivered to other compartments of the host cell. Although many plastid genes are hosted and expressed in the nucleus, plastids do still retain some essential genes as well as the

ability to replicate and transcribe their genomes. This feature of plastids was highlighted in a recent study of chloroplast transcription in which the authors determined that the complete plastid genome was transcribed, pointing to the importance of post-transcriptional processing and regulation (*48*). Adding to the complexity of the plastid proteome, it is also possible for non-cyanobacterial prokaryotic genes to be acquired by the eukaryotic host and subsequently for the translated protein product to be targeted to the plastid (*49*). One example is polyphenol oxidase, which exhibits a strong deleterious effect on plant growth when it is not targeted to the plastid, suggesting a selective pressure for targeting this protein to the plastid. In addition to the genomic and proteomic integration, a recent review of data led to the conclusion that nonpolar metabolites are directly exchanged between the chloroplast and the endoplasmic reticulum by means of membrane hemifusion (*50*). The consequence of the ability for chloroplasts to exchange nonpolar metabolites by physical membrane contact is that the nonpolar metabolic pathways can evolve independently of transporter proteins. This is in contrast to polar metabolites, for which there are many known transporter proteins that are required to mediate exchange between the chloroplast stroma and the cytoplasm. Beyond the exchange of genes and the transport of metabolites between the chloroplast and the host cell, there are other important and dynamic interactions. For example, a recent review highlighted the importance of plastid translation in generating a currently unknown retrograde signal of some kind that provides feedback to the host cell nucleus and regulates gene expression therein (*51*). This interaction has a very strong influence on plant developmental and morphological phenotype, and presents interesting areas for further research.

Plastid genomes are proving immensely useful for the resolution of phylogenetic and taxonomic hierarchies. The Chlorophyta (green algae) are particularly difficult to classify due to the enormous amount of morphological diversity within the clade. Chloroplast genomes of green

algae were recently utilized in an attempt to discern the phylogenetic relationships of the six nominal classes within the Chlorophyta (*52*). The authors concluded that the class Chlorophyceae is monophyletic, whereas the classes Ulvophyceae, Prasinophyceae, and Trebouxiouphyceae are non-monophyletic. One chloroplast phylogenomics analysis focused on the Trebouxiophyceae class, which contains a large sample of morphologically and ecologically diverse species (*53*). The authors concluded that the Trebouxiophyceae class is non-monophyletic, in agreement with results from other studies. Another recent phylogenomic analysis provided evidence of two clades, the Prasinococcales and the Palmophyllales, that together form the deepest-branching clade of Chlorophyta (*54*). The authors deem this new grouping a class within the phylum Chlorophyta called the Palmophyllophyceae. A recent review discussed contemporary efforts at resolving the taxonomic organization of Chlorophyta through chloroplast phylogenomic analyses (*55*). The authors highlighted the limits and biases of such analyses and considered options to improve the accuracy of phylogenomic results. Two major takeaways are the need for better evolutionary models and increased taxon sampling. Moreover, it is important to consider that chloroplast genomes separated by speciation are subject to differential selective pressures and still undergo evolution. Analysis of 38 chloroplast genomes from the green algal classes Trebouxiophyceae and Pedinophyceae revealed the degree of diversity that has accrued in the plastid genomes through divergent evolution (*56*). The authors found that while a number of genes were conserved in all the chloroplasts, there were substantial losses, expansions, contractions, and rearrangements. They also found evidence of putative horizontal gene transfers. Thus chloroplast phylogenomics faces important challenges to address in order to ensure the validity of the results obtained through this approach.

1.1.3 Functional Diversification in Viridiplantae

Our understanding of the evolutionary history of algae has been transformed by the development of next-generation sequencing technologies. Ten years ago, only a handful of plant genomes were available for comparative analyses (*57*). Nonetheless, the importance of obtaining genome sequences was very clear at the time and since then many more genomes have been sequenced and made publicly available. This data is helping to answer key questions about the evolution of important plant traits, including photosynthesis, multicellularity, morphological development, and differentiated tissues. With the ability to peer into the complete genomes of a wide range of taxa, the dots of evidence can be connected and the evolutionary web grows more complete with each added genome. In a recent review, the authors survey our current understanding of algal evolution, discuss the only other known primary endosymbiotic event, and speculate as to why the occurrence of primary endosymbiosis is so rare, whereas secondary and tertiary endosymbiosis are more common events (*58*). When studying the evolution of photosynthetic eukaryotes, it is important to consider the environmental context, in particular the atmospheric conditions. In a recent review, the authors discuss the dynamics of atmospheric carbon dioxide concentrations and attempt to provide context for the evolution of algae (*59*). In summary, all known oxygenic organisms utilize the ribulose bisphosphate carboxylase-oxygenase (RuBisCO) pathway (i.e. Calvin cycle) to fix carbon dioxide into organic compounds. As the atmospheric concentration of oxygen increased, the oxygenase activity of RuBisCO also increased, reducing the efficiency of carbon fixation. This resulted in a selective pressure for the evolution of carbon-concentrating mechanisms (CCMs), which are ubiquitous in cyanobacteria. However, with the diversification of photosynthetic eukaryotes, some species lost the CCM and developed alternative pathways for efficient delivery of carbon to RuBisCO. Despite the wide variation in

genetic traits across the green plant lineage, some characteristics evolve convergently with remarkable consistency, such as C4 photosynthetic carbon fixation, which has at least sixty independent origins (*60*). Crassulacean acid metabolism (CAM) is another example of convergent evolution in the photosynthetic carbon fixation pathway and has been well studied in the pineapple genome (*61*). It was determined that reconfiguration of preexisting C3 pathways and not gene acquisition led to the emergence of CAM in this species. However, there are many examples of innovation beyond photosynthesis and carbon fixation.

As the basal Viridiplantae species radiated into the environment, they developed greater complexity and a diversity of functional traits. A recent review thoroughly discussed the current understanding of the evolution of green algae and land plants (*62*). The authors emphasize the important role that whole-genome data has played in the improvement of evolutionary models describing Viridiplantae. They also discuss the evolution of organellar genomes and multicellular morphologies. Finally, they highlight the need for a larger sampling of taxa to improve the resolution of early diversification in this ancient lineage. An accurate reconstruction of the evolutionary progression in Viridiplantae is important for illuminating the molecular mechanisms underpinning the diverse morphological and functional features of the various species. Another review, from just before the modern genomic era, attempted to reconstruct the evolutionary progression of Viridiplantae via synthesis of then current phylogenetic data and the fossil record (*63*). While it is commonly accepted that land plants evolved from early-branching freshwater green algae in the Streptophyta clade, the authors argue that both the Streptophyta and Chlorophyta evolved from a common marine ancestor. They further suggest that the Chlorophyceae and Trebouxiophyceae groups within the Chlorophyta adapted from a marine environment to a freshwater environment. This is in stark contrast to the recent data supporting the evolution of

Archaeplastida in a freshwater environment. One fairly recent review summarized the current views on the evolution of the green plant lineage and places the origin of this lineage as early as 1.5 Ga, in agreement with other estimates (*64*). Subsequently, two major branches formed, the Chlorophyta and the Streptophyta, dominating mostly marine and strictly freshwater habitats, respectively. Land plants emerged from the streptophyte algae approximately 476 Ma. The authors discuss the difficulties of phylogenetic reconstruction and attempt to line up this data with observations made from the sparse record of fossils. A more recent review thoroughly discusses the challenges of phylogenomics and the emerging picture of phylogenetic relationships that led to the evolution of land plants (*65*).

The evolution of multicellular structures and differential tissue types enabled the Viridiplantae to assume the vast morphological architectures that are currently observed. A recent review surveyed the development of multicellularity in the Volvocales lineage of green algae (*66*). In summary, multicellularity is thought to have evolved independently at least 25 times, occurring relatively recently in the Volvocales (approximately 200 Ma) as compared to some other lineages like animals and land plants (0.65 – 1 Ga). A significant benefit of using Volvocales to study the emergence of multicellularity is that it is a very simple system, especially in comparison to animals and land plants. The major process driving the emergence of multicellularity is thought to be co-option of existing genes for new functions, although *de novo* gene evolution may play a role (but this is not well understood). The first step in the evolution of multicellularity in the Volvocales, and possibly other lineages, indeed appears to be the co-option of retinoblastoma (RB) and its regulation of the cell division cycle, as revealed by genomic analysis of *Gonium pectorale* (*67*). Most interestingly, when the authors expressed the *Gonium* RB in *Chlamydomonas reinhartdii*, it induced the formation of colonial structures, demonstrating biochemical function. In contrast, the

traditional view has been that the evolution of multicellularity was dependent on the rewiring of transcription networks, possibly through the acquisition and expansion of transcription factor families. Transcription factors govern the degree and timing of gene expression, and are critical regulators of all cellular processes. An analysis of transcription factor diversity across eukaryotes reveals that the repertoire of transcription factors increases with organismal complexity (*68*). This data suggests that transcription factors play a significant role in the transition from unicellular to multicellular life. However, this paper does not demonstrate a direct mechanistic function of transcription factors in generating multicellularity. Analysis of transcription factor families in microalgae revealed the presence of cyanobacterial and fungal transcription factors in the algal nuclear genomes (*69*). This data provides further evidence of endosymbiotic and horizontal gene transfer, and further illustrates the complex evolutionary processes impacting transcriptional regulation. Although transcription factors may or may not play a direct role in the emergence of multicellularity, it has been clearly shown that they play an important role in tissue specificity. The control of differential cellular male and female mating types in *Volvox carteri* is governed by a single transcription factor (*70*). Misexpression of the transcription factor led to the uncoupling of sex determination from sex chromosome identity, confirming its role. The data also demonstrate evidence for gender-specific adaptions in the male and female loci in *V. carteri*.

The transition from simple single-celled green algae to complex multicellular land plants is of substantial interest. Gaining insight into the process underlying this transition requires a deeper understanding of the genetics that span the transition. Near the base of the green lineage are the Prasinophyceae, which include *Bathycoccus prasinos* and *Micromonas pusilla*, small green algae with genomes of only 15 Mbp and 22 Mbp respectively. *Bathycoccus* species synthesize scales, which cover the surfaces of the cells, forming a protective layer. By comparing the genome

of *B. prasinos* with other green algae, it was revealed that four gene families were highly expanded in this species, coding for enzymes with functions including sialyltransferases, sialidases, ankyrin repeats, and zinc ion-binding. The authors hypothesized that these genes are involved in scale biosynthesis (*71*). The genomes of two *Micromonas* isolates were sequenced and analyzed, revealing basal Viridiplantae traits, such as components of photosynthesis and peptidoglycan biosynthesis which have been selectively retained or lost in various lineages of higher plants (*72*). A key takeaway from this work is that even within the genus *Micromonas*, there is substantial genomic variation between species, highlighting the importance of increasing the diversity of taxon sampling for comparative studies. On the other end of the Viridiplantae spectrum, the moss *Physcomitrella patens*, the liverwort *Marchantia polymorpha*, and the lycophyte *Selaginella moellendorfii*, are species considered representative of the earliest land plants. Analysis of the light-harvesting antenna complexes encoded in the *P. patens* genome has yielded some insight into the characteristics of photosynthesis as it moved from an aquatic to a terrestrial environment (*73*). The genome of *M. polymorpha* revealed innovations in biochemical pathways, phytohormone signaling pathways (especially auxin), expansions of other signaling pathways, and diversification in some transcription factor families (*74*). The genome of *S. moellendorfii* showed further innovations in post-transcriptional gene regulation, small RNA regulation of repeats, RNA-editing of organellar genes, and a lack of small interfering RNA mechanisms (*75*). Recently, a comparative genomics analysis of green algae and land plants gave insight into the evolution of traits required for colonization of land (*76*). In particular, it was found that the ability to form symbiotic associations is a critical trait that expanded in early land plants. This can occur through evolutionary mechanisms such as gene duplication and neofunctionalization, and potentially horizontal gene transfer. The authors conclude that since some of these symbiosis pathways are

present in algae in a reduced form, the algal precursors of land plants were preadapted for the development of more complex symbioses that aided the transition to land. With more genome sequencing projects, the evolutionary progression from algae to land plants and the roles of gene evolution and genetic exchange will come into sharper focus. Today there are approximately 278 Viridiplantae genomes available from databases hosted by the National Center for Biotechnology Information. The availability of large amounts of complete genome sequence data for plants has enabled efforts to determine ancestral genome content. This procedure is based on three steps: 1) identification of orthologs and paralogs, 2) identification of collinear genes and syntenic blocks, and 3) inferred reconstruction of the ancestral genome (*77*). This procedure theoretically enables researchers to study the course of genome evolution directly from ancestral karyotypes to current genomic structures. This ancestral genome reconstruction method was applied to flowering plants (i.e. angiosperms), revealing a potential pool of 22,899 ancestral genes, which are conserved in present-day plants (*78*). Based on these genes, the authors estimated that the most recent common ancestor of flowering plants emerged approximately 214 Ma, earlier than is suggested by the fossil record.

With a wide range of sequenced Viridiplantae genomes now available, accompanied by detailed analyses of the functional contents, the processes enabling genome expansion and evolution can be further investigated. The duplication of genes through mechanisms involving transposable elements, as well as other mechanisms such as whole genome duplication, is a very important phenomenon, because it is well established that gene duplication is a driver of plant evolution (*79*). To gain insight into evolutionary processes in action, genetic variation within a species was observed recently in a population genomics analysis of *Populus trichocarpa* (black cottonwood) across a wide geographical range (*80*). The authors found 397 genomic regions that

showed evidence of natural selection and found clues about the roles of duplicated genes in adaptive trait variation. Transposable elements can lead to the duplication, mobilization, or recombination of nearby genes and gene fragments, contributing to the emergence of paralogs that can undergo neofunctionalization. A recent review discusses the role of transposable elements in the evolution of plant genomes (*81*). These self-replicating elements are a major source of genetic mutation and also exert influence on the expression levels of nearby genes. Another review on transposable elements, slightly older, presents more discussion on the mechanisms of action in affecting gene regulation (*82*). Additionally, the authors discuss epigenetic factors that control the expression and propagation of transposable elements. The epigenetic aspect is increasingly important as scientists begin to develop a better understanding of the role that epigenetic mechanisms play in the expression and evolution of whole genomes (*83*). One review of transposable elements focuses on the functional impact they have on genes and the resulting plant phenotypes (*84*). This perspective is important because the core mission of biology is to connect genotype and phenotype through a mechanistic understanding of genome organization and interpretation. Another interesting perspective on the evolution of transposable elements and epigenetic silencing mechanisms posits a reversal of traditional assumptions (*85*). Whereas one may view the evolution of silencing mechanisms as a response to transposable elements, the author suggests that the expansion of transposable elements throughout genomes is in fact a consequence of the evolution of silencing mechanisms, having enabled the genetic management of these jumping genes. However, transposable elements do have the ability to massively inflate the size of a genome, such as that of *Picea abies* (Norway spruce), which has a genome size of approximately 20 Gbp (*86*). Despite this very large genome size, it only contains an estimated 28,354 genes, approximately the same number as found in *Arabidopsis thaliana*, which has a

genome size of approximately 122 Mbp. Given the immense difference in morphology between these two species, it is interesting that the number of genes is not very different, and begs the question of what genetic factors contribute to their differences. Beyond the function of transposable elements, other types of DNA elements are important for genome evolution. Tandem repeats (TRs) are DNA sequence elements that consist of two or more repeating motifs with variable unit sizes (e.g. microsatellites: 1-6 bp, minisatellites: 10-100 bp). Genome-wide analysis of TRs in Viridiplantae genomes revealed a wide range of TR abundance, with no correlation to genome size. The authors argue that the TRs are nonrandomly distributed in genes and indicate that these elements may play a role in transcriptional or translational regulation (*87*). Although the concepts of selective pressure and evolution have been in existence for approximately 160 years since Darwin first published them in his seminal work *On the Origin of Species* in 1859, the mechanisms of genomic evolution have only recently come into view. Much work remains to be done in this field in order to more fully understand these processes and the biological functions that emerge as a consequence.

1.2 History of *Botryococccus braunii*

Botryococcus braunii is a colony-forming green microalga (Figure 1). The colonial extracellular matrix (ECM) consists of crosslinked hydrocarbon polymers, holding together clusters of single cells. Each of the cells is embedded in the ECM near the surface of the colony, with the apical surface of the cell exposed. The surface of the colony is covered with a polysaccharide sheath, the precursors of which are synthesized in the cells and secreted to the surface, where they are polymerized by an unknown mechanism. The colonial ECM is filled with liquid hydrocarbons, which are synthesized and temporarily stored in the cells, prior to secretion into the ECM by an unknown mechanism. When a sufficiently large amount of oil has accumulated in the ECM, the *B. braunii* colonies become buoyant.

There are three major types of *B. braunii* (called "races"), which are chemically distinct but morphologically very similar. The races are distinguished by the types of hydrocarbons that they synthesize and store in the ECM (Figure 2). The A race produces fatty-acid derived alkenes with two or three degrees of unsaturation. The B race produces triterpenoid hydrocarbons called botryococcenes, as well as polymethylated squalenes. The L race produces a tetraterpenoid hydrocarbon called lycopadiene. The B race in particular has received attention due to the readiness with which botryococcene can be transformed into distillate fuels (Figure 3).

The story of *B. braunii* begins in the mid-19th century and comes into focus very slowly until the advent of the 21st century, at which point the rate of knowledge accumulation rapidly increased. The following section comprehensively and chronologically covers the process of discovery that has resulted in the current understanding of this interesting organism. The details of its physiology will be explained as they were discovered, as well as who discovered them.

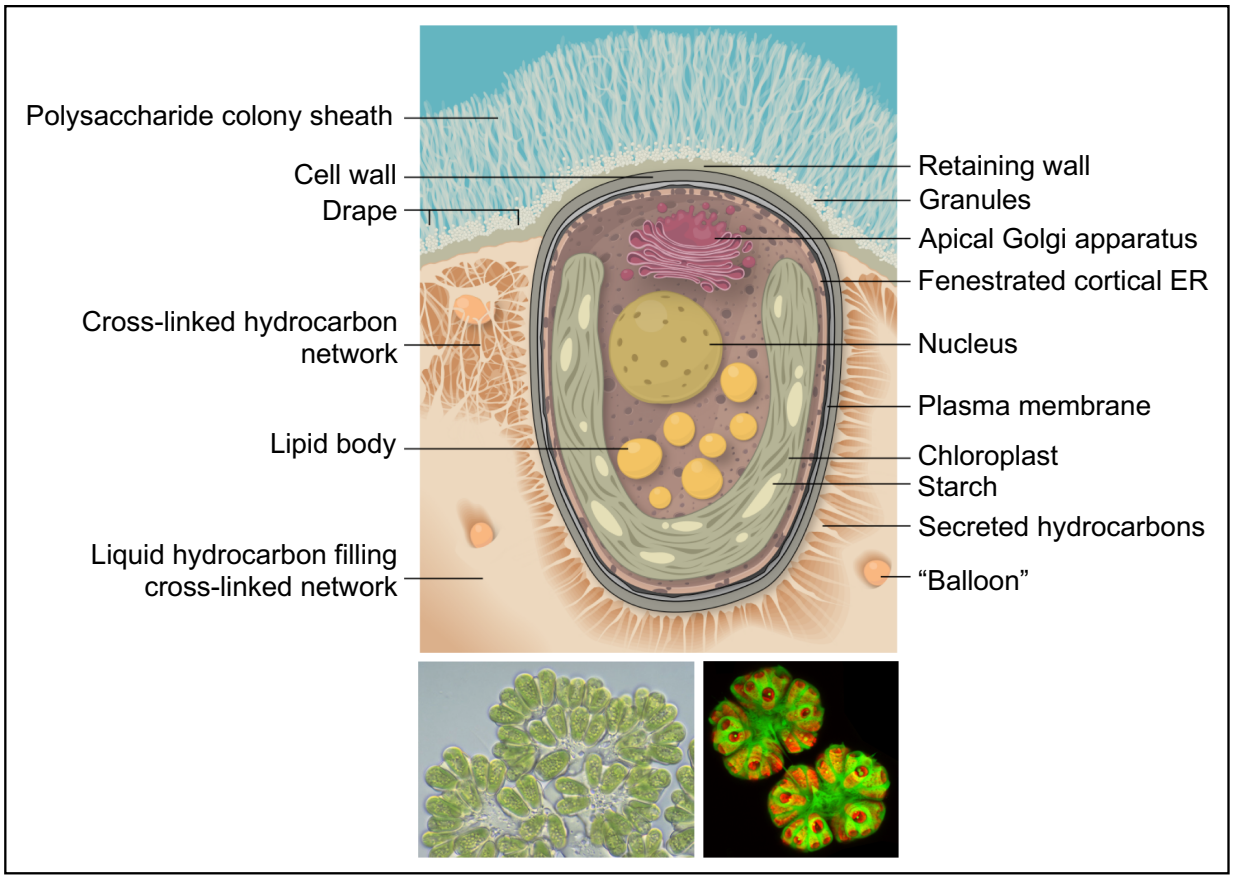


Figure 1. Model of *Botryococcus braunii* cell morphology and colony structure. This figure shows a false-color image of two *B. braunii* colonies (top left), a white-light microscope image of cells in a colony (bottom left), and a cartoon model of one cell embedded in a colony (right). The cartoon model shows the details of *B. braunii* cell biology, with various organelles and cellular components labeled. This model summarizes the current knowledge of *B. braunii* morphology. Figure adapted from Weiss et al., 2012 *Eukaryotic Cell* **11:**1424-1440.

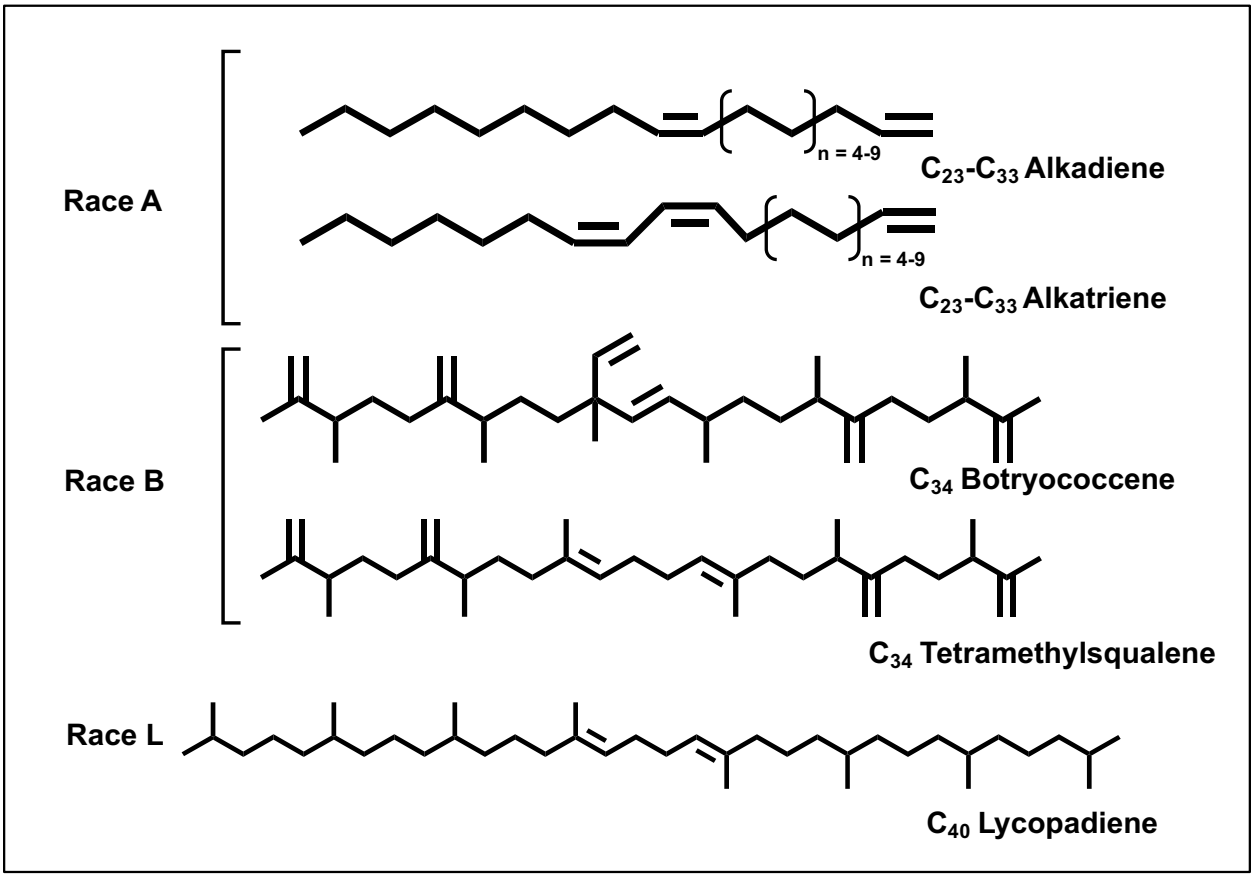


Figure 2. Molecular structures of hydrocarbons specifically synthesized by each race of *Botryococcus braunii*. The different races of *B. braunii* are distinguishable by the types of hydrocarbons that accumulate in the colonial extracellular matrix. The B and L races produce terpenoid hydrocarbons while the A race produces hydrocarbons derived from fatty acids.

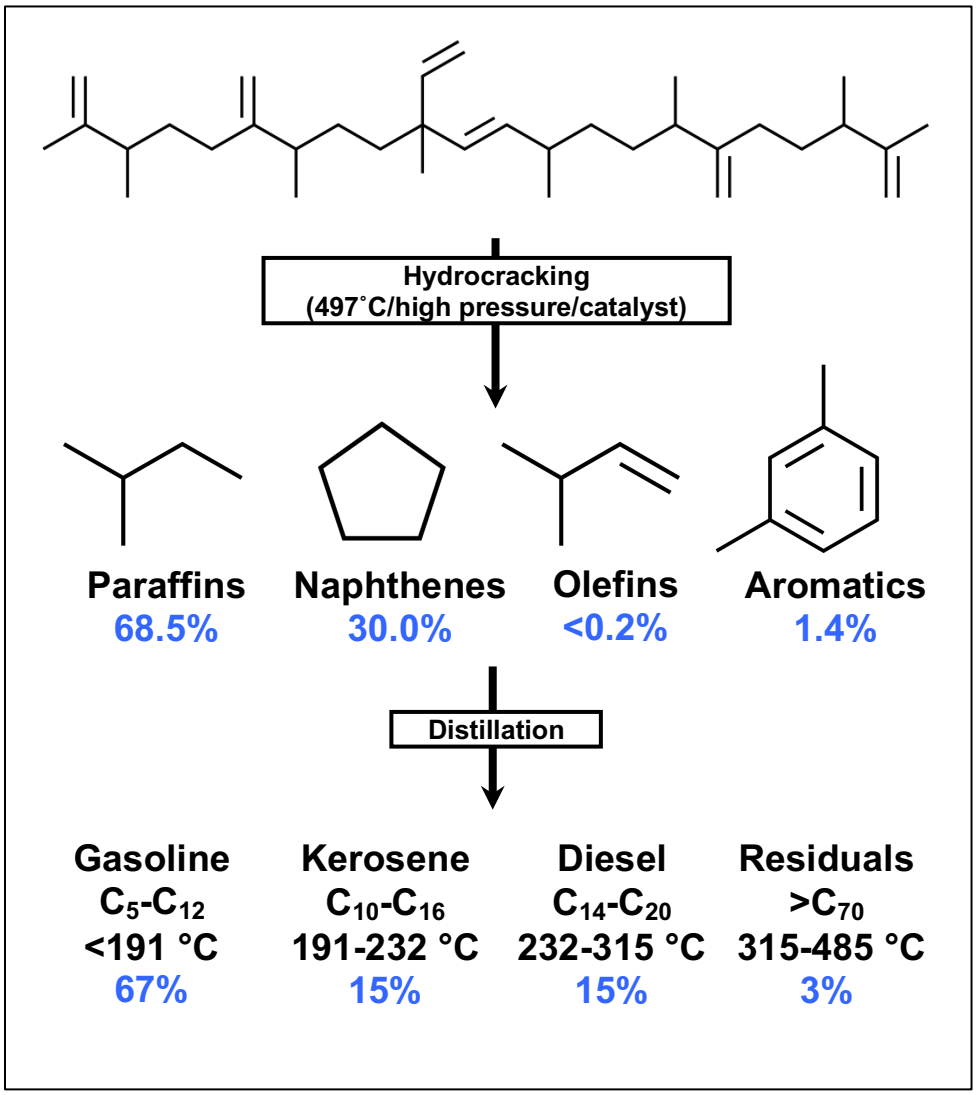


Figure 3. Processing and distillation of botryococcene yields petroleum-like fractions. The botryococcene molecules can be cracked and distilled with conventional techniques used to process petroleum. The molecules yielded from processing botryococcene closely resemble those obtained from petroleum. This shows that hydrocarbons from *B. braunii* can be utilized as a direct replacement for petroleum. Figure adapted from Hillen et al., 1982 *Biotechnol Bioeng* **24:**193-205.

1.2.1 The 20th Century

Friedrich Traugott Kutzing, a German pharmacist, botanist, and phycologist, at the University of Giessen, published the first known description of *B. braunii* in 1849 as part of his seminal work *Species Algarum* (Figure 4). This massive tome, written in Latin, contains nearly one thousand pages and describes approximately six thousand species of algae, constituting the most comprehensive classification of algal species at the time. However, it is possible that there was an earlier description of *B. braunii* in 1835 by Carl Adolph Agardh, a Swedish botanist at Lund University, in his book *Icones Algarum Europaearum*. In this work, there are drawings of a species named *Palmella botryoides*, which roughly appear to resemble colonies of *B. braunii*, but the lack of detail makes it very difficult to know for certain (Figure 5). By 1896, the specific knowledge of *B. braunii* had grown sufficiently to warrant a review, published by Robert Chodat, a Swiss botanist and phycologist, at the University of Geneva (*88*). Unfortunately, this publication is written in French and has not yet been translated into English, making it difficult to provide a more complete summary of the knowledge presented therein. Four decades later, in 1936, Kathleen Blackburn, a British botanist at the University of Durham, published a remarkably detailed analysis of *B. braunii* colonial and cellular morphology (*89*). The careful application of various dyes and meticulous observation by microscope enabled her to make incredibly accurate drawings of the alga (Figure 6). In addition to elegantly summarizing the other work done on *B. braunii* in the early 1900s, Blackburn substantially advanced the state of knowledge about this species. There would not be a more detailed study of *B. braunii* morphology for nearly 50 years. The opportunity to study any species of algae in greater detail is dependent upon the ability to cultivate that species in the laboratory. In 1942, Chu, a botanist at the University of London, conducted a fairly comprehensive study on the impact of media mineral composition on the growth of several algae,

including *B. braunii* (*90*). Prior to this development, all studies of *B. braunii* were dependent upon the collection of samples from natural environments. With the media formulations compiled by Chu, researchers were able to maintain collections of algal species isolated from environmental samples and grow them in standardized laboratory conditions.

Much of the early scientific debate about *B. braunii* centered on the question of taxonomic classification, based on morphological characteristics, with competing assignments made to the Xanthophyceae and the Chlorophyceae. Although both of these assignments would later prove erroneous, Belcher and Fogg, botanists at University College, London, argued in 1955 that *B. braunii* belonged to the Chlorophyceae on the basis of its chlorophyll components (*91*). However, until now the major area of study concerning *B. braunii* had less to do with the living species and more to do with the identification of fossils. Since the late 1800s, researchers had recognized the importance of *B. braunii* in the formation of energy-dense sediments like coal and shale, comprehensively summarized in 1955 by Traverse, a scientist at the US Bureau of Mines (*92*). Microfossils identified as remnants of *B. braunii* had been found in sediments spanning the Phanerozoic Eon (541 – 0 Ma). Although the primary interest in *B. braunii* was driven by its unique production of hydrocarbons and the role they played in the formation of coal and shale, the molecular identity of these hydrocarbons was largely unknown. In 1968, Gelpi *et al* at the University of Houston applied gas chromatography and mass spectrometry for the first time to oils extracted from *B. braunii* and reported their findings in the prestigious journal *Science* (*93*). They identified the hydrocarbons as aliphatic dienes and trienes with the dominant fraction consisting of 27-, 29-, and 31-carbon chains, although they did not determine the positions of the double bonds (Figure 7). Just four months later, Maxwell *et al* from the University of Glasgow reported in *Phytochemistry* the detailed structures of hydrocarbons they isolated from *B. braunii* (*94*). In

addition to gas chromatography and mass spectrometry, they also applied for the first time nuclear magnetic resonance spectroscopy, infrared spectroscopy, and chemical hydrogenation and reduction reactions to resolve the structures. They termed this novel class of hydrocarbons the botryococcenes. The reports of both unbranched aliphatic alkenes and botryococcenes presented a conundrum as to why these different hydrocarbons were separately identified from samples of the same species. In 1969, Brown and Knights, also at the University of Glasgow, proposed the existence of "physiological states" that determine the hydrocarbon contents of *B. braunii* (*95*). Principally, they hypothesized that "active state" colonies produce the aliphatic alkenes, while "resting state" colonies produce botryococcenes, and that state transitions were a function of growth conditions. While this hypothesis would eventually be disproven, it held sway in the field for over a decade.

The advent of the nuclear age brought with it the invention of carbon radioisotopes that enabled researchers to study biosynthetic pathways in greater detail. The first application of this radiolabeling approach to the study of *B. braunii* occurred in 1977, when Murray and Thomson, at Torry Research Station in Scotland, grew *B. braunii* cultures with ^{14}C-labeled sodium carbonate as a carbon source (*96*). While their study was quite modest and merely confirmed the biosynthesis of unsaturated hydrocarbons by *B. braunii* (and not by any contaminating organisms), this technique would transform the study of *B. braunii* in the next decade. In early 1980, the first paper emerged from a group at the Centre National de la Recherche Scientifique (CNRS) in France (*97*). This group, composed mainly of Claire Berkaloff, Eliette Casadevall, Sylvie Derenne, Claude Largeau, Pierre Metzger, and Joelle Templier, would be the dominant force driving forward research of *B. braunii* for nearly two decades. Their first paper was focused on the localization of hydrocarbon storage in *B. braunii*, utilizing Raman spectroscopy and electron microscopy to

demonstrate that the alga mainly accumulates hydrocarbons in the colonial extracellular matrix, with only a small fraction of the total hydrocarbons found inside the cells (Figure 8). Later that year, the same group conducted the second radiolabeling analysis of metabolism in *B. braunii* using ^{14}C-labeled palmitic acid in a feeding experiment (*98*). They observed incorporation of the radiolabel into the hydrocarbon fraction, with the majority of radiolabel accumulating in the extracellular pool. Although they erroneously concluded that the alga does not have an active excretory process, arguing instead for extracellular biosynthesis of hydrocarbons, they did correctly conclude that it does not catabolize the hydrocarbons. The year 1980 also saw an important report on a natural bloom of *B. braunii*, occurring in the Darwin River Reservoir in Australia (*99*). The authors, Wake and Hillen, monitored the ecological conditions occurring during the bloom and tried to develop an understanding of the biotic and abiotic factors underpinning the bloom phenomenon. Furthermore, they measured the hydrocarbon contents of the bloom and studied biomass desiccation in relation to the deposition of fossilized algae and the formation of oil shale. Two years after their publication on the bloom in Darwin River Reservoir, Wake and Hillen published a paper demonstrating the production of fuel oils from *B. braunii* hydrocarbons via hydrocracking (*100*) (Figure 9). The oil crisis of 1973 had clearly shown the strategic vulnerability of petroleum dependency and provoked substantial interest in renewable fuel technologies. Now, *B. braunii* was beginning to look like a promising candidate for commercial production as an energy crop, at least from a technical, if not yet an economical, perspective.

With the body of literature describing *B. braunii* growing slowly since the early 1900s, the time was ripe for a review. In 1983, Fred Wolf, who earned his doctorate in 1981 studying *B. braunii* at Texas A&M University, published a review in the journal *Applied Biochemistry and*

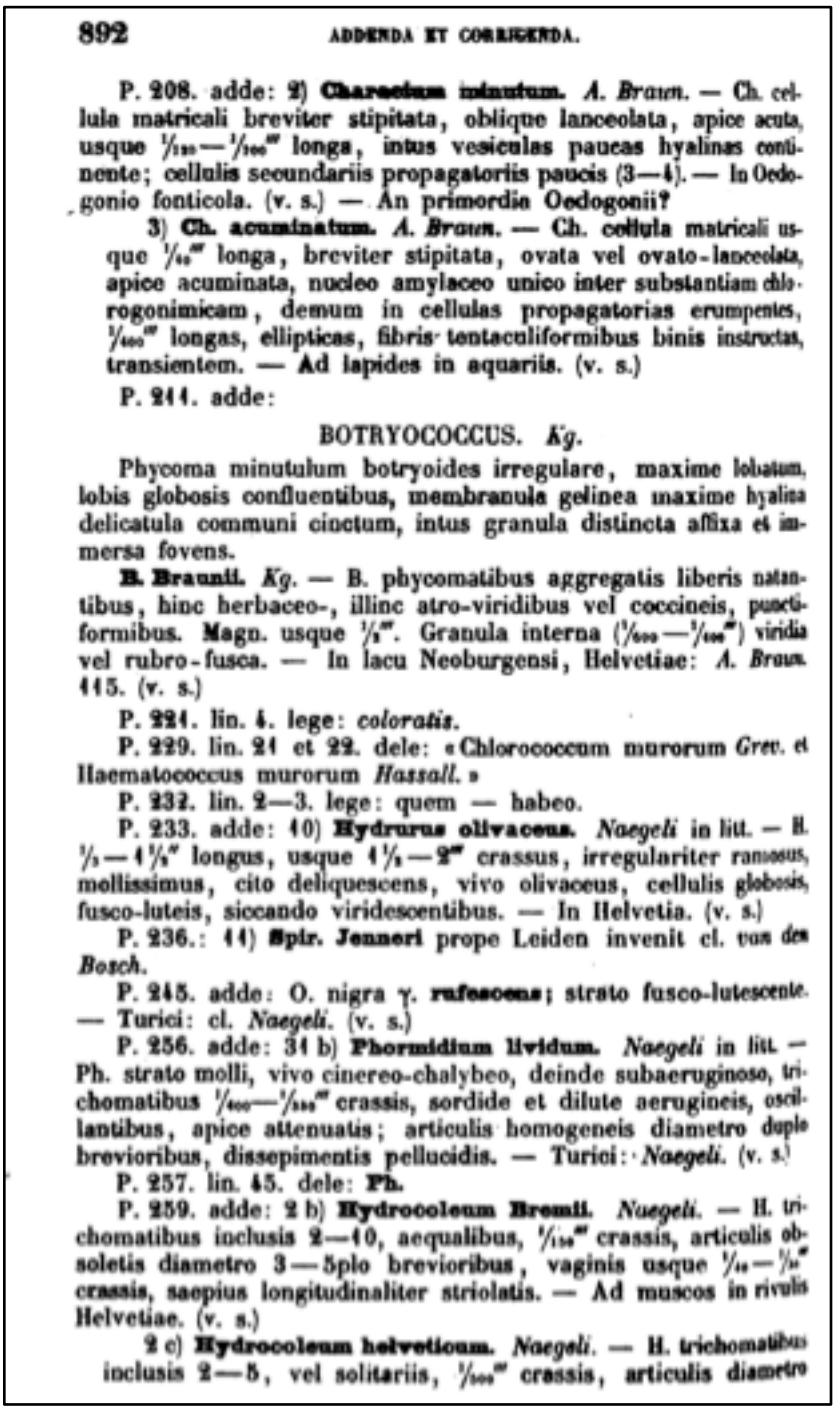
Biotechnology (*101*). In summarizing the state of knowledge, Wolf did an exceptional job of pointing out exactly how little was known about the alga, positing many important questions, and casting doubt on the near-term possibility of *B. braunii* as a commercially viable source of renewable fuel. One of the most important questions posited by Wolf was with respect to what factors control the transition from "active" to "resting" state. The first hint of a solution to the "active and resting state" problem appeared in 1984, when Berkaloff *et al* conducted a comparative analysis of a wide variety of *B. braunii* strains (*102*). They observed no correlation between hydrocarbon content and colony structure throughout various stages of growth across all the strains. However, they did observe that botryococcenes were never found in occurrence with dienic hydrocarbons, and concluded that there may be distinct ecotypes of *B. braunii* that synthesize different hydrocarbons. The following year, Metzger *et al* confirmed this conclusion, ending the era of the "active and resting state" hypothesis (*103*). They established the current paradigm of *B. braunii* "races" that are morphologically very similar, but are distinguished by their hydrocarbon contents. Irrespective of growth stage and conditions, the "A race" produces unbranched alkadienes and trienes, and the "B race" produces various polymethylated triterpenes (e.g. botryococcenes). Subsequently, in 1987, Metzger and Casadevall discovered strains of *B. braunii* that uniquely produce a tetraterpenoid called lycopadiene, establishing a new "L race" of the species (*104*).

Although it was already well known that *B. braunii* is a component of many organic sedimentary deposits throughout the Phanerozoic Eon, in 1989, Glikson *et al* in Australia found evidence of even older fossils (*105*). Utilizing the detailed ultrastructural analyses of *B. braunii* that had recently become available as a reference, they definitively identified algal remnants with transmission electron microscopy in sediments dated to the Neoproterozoic Era (1,000 – 541 Ma).

Furthermore, by studying geographically diverse petroleum source rocks, and observing remains of *B. braunii* in many of them, the authors concluded that this alga has been a major contributor of organic matter to petroleum formations throughout geological time. By the end of the 1980s, researchers had generated substantial amounts of new knowledge about *B. braunii*, creating the conditions for another review to be published. In 1992, Dorothy Guy-Ohlson at the Swedish Museum of Natural History wrote a review, focused on *B. braunii* from the perspective of paleobiology and the utilization of algal fossils as indicators of ancient environmental conditions (*106*). However, she completely neglected to include any of the major advances of the 1980s from the CNRS group and others, and referred erroneously to the clearly disproven "active and resting state" hypothesis. As a result, this review provides almost no value to anyone seeking a greater understanding of contemporary knowledge about *B. braunii* physiology. The one particularly interesting piece of information from her review is an illustration of the developmental stages in the life cycle of *B. braunii* (Figure 10). However, the processes conveyed in the illustration are not strongly supported with scientific citations.

While much of the early research on *B. braunii* consisted of determining its place in the taxonomic hierarchy, the subject came under scrutiny again in 1995, when Sawayama *et al* from Japan conducted a phylogenetic analysis (*107*). Utilizing the tools of polymerase chain reaction and DNA sequencing, they determined the sequence of a small subunit ribosomal RNA amplified from *B. braunii*. They compared this sequence to those of 13 other species and constructed a phylogenetic tree using parsimony maximization. Based on this data, they concluded that *B. braunii* belongs in the Chlorophyceae and that its closest relatives with available sequences were *Characium vacuolatum* and *Dunaliella parva*. As the 20th Century drew to a close, improvements in technology continued to stoke new developments in *B. braunii* research. In 1998, Beakes and

Cleary from Australia used laser scanning confocal microscopy to image chloroplast autofluorescence and lipophilic dye fluorescence in living *B. braunii* samples for the first time (*108*). Their work yielded important morphological insights and opened the door to obtaining a deeper understanding of processes such as hydrocarbon secretion and colony formation (Figure 11). The final publication on *B. braunii* of the millennium was a very significant milestone: the first gene cloned from *B. braunii* and heterologously expressed in the bacteria *Escherichia coli* (*109*). The gene encoded the malate dehydrogenase enzyme, and when expressed in *E. coli*, the cells showed a substantial increase in the conversion of oxaloacetate to malate, demonstrating biochemical activity. This apparently small step forward foreshadows giant leaps in deciphering metabolic pathways and enzymatic mechanisms in *B. braunii*.



892 ADDENDA ET CORRIGENDA.

P. 208. adde: 2) **Characium minutum.** *A. Braun.* — Ch. cellula matricali breviter stipitata, oblique lanceolata, apice acuta, usque $^1/_{130}$—$^1/_{100}$‴ longa, intus vesiculas paucas hyalinas continente; cellulis secundariis propagatoriis paucis (3—4). — In Oedogonio fonticola. (v. s.) — An primordia Oedogonii?

3) **Ch. acuminatum.** *A. Braun.* — Ch. cellula matricali usque $^1/_{40}$‴ longa, breviter stipitata, ovata vel ovato-lanceolata, apice acuminata, nucleo amylaceo unico inter substantiam chlorogonimicam, demum in cellulas propagatorias erumpentes, $^1/_{400}$‴ longas, ellipticas, fibris tentaculiformibus binis instructas, transientem. — **Ad lapides in aquariis.** (v. s.)

P. 214. adde:

BOTRYOCOCCUS. *Kg.*

Phycoma minutulum botryoides irregulare, maxime lobatum, lobis globosis confluentibus, membranula gelinea maxime hyalina delicatula communi cinctum, intus granula distincta affixa et immersa fovens.

B. Braunii. *Kg.* — B. phycomatibus aggregatis liberis natantibus, hinc herbaceo-, illinc atro-viridibus vel coccineis, punctiformibus. Magn. usque $^1/_3$‴. Granula interna ($^1/_{500}$—$^1/_{100}$‴) viridia vel rubro-fusca. — In lacu Neoburgensi, Helvetiae: *A. Braun.* 115. (v. s.)

P. 221. lin. 4. lege: *coloratis.*

P. 229. lin. 21 et 22. dele: «Chlorococcum murorum *Grev.* et Haematococcus murorum *Hassall.*»

P. 232. lin. 2—3. lege: quem — habeo.

P. 233. adde: 10) **Hydrurus olivaceus.** *Naegeli* in litt. — H. $^1/_3$—$1^1/_2$″ longus, usque $1^1/_2$—2‴ crassus, irregulariter ramosus, mollissimus, cito deliquescens, vivo olivaceus, cellulis globosis, fusco-luteis, siccando viridescentibus. — In Helvetia. (v. s.)

P. 236.: 11) **Spir. Jenneri** prope Leiden invenit cl. van den *Bosch.*

P. 245. adde: O. nigra γ. **rufescens**; strato fusco-lutescente. — Turici: cl. *Naegeli.* (v. s.)

P. 256. adde: 31 b) **Phormidium lividum.** *Naegeli* in litt. — Ph. strato molli, vivo cinereo-chalybeo, deinde subaeruginoso, trichomatibus $^1/_{400}$—$^1/_{333}$‴ crassis, sordide et dilute aerugineis, oscillantibus, apice attenuatis; articulis homogeneis diametro duplo brevioribus, dissepimentis pellucidis. — Turici: *Naegeli.* (v. s.)

P. 257. lin. 15. dele: **Ph.**

P. 259. adde: 2 b) **Hydrocoleum Bremii.** *Naegeli.* — H. trichomatibus inclusis 2—10, aequalibus, $^1/_{130}$‴ crassis, articulis obsoletis diametro 3—5plo brevioribus, vaginis usque $^1/_{10}$—$^1/_{12}$‴ crassis, saepius longitudinaliter striolatis. — Ad muscos in rivulis Helvetiae. (v. s.)

2 c) **Hydrocoleum helveticum.** *Naegeli.* — H. trichomatibus inclusis 2—5, vel solitariis, $^1/_{500}$‴ crassis, articulis diametro

Figure 4. First known description of *Botryococcus braunii.* This figure shows the first known written record of *B. braunii*, from the Germany botanist Friedrich Kutzing in 1849. Figure reprinted from Kützing, 1849 *Species Algarum* FA Brockhaus, Leipzig.

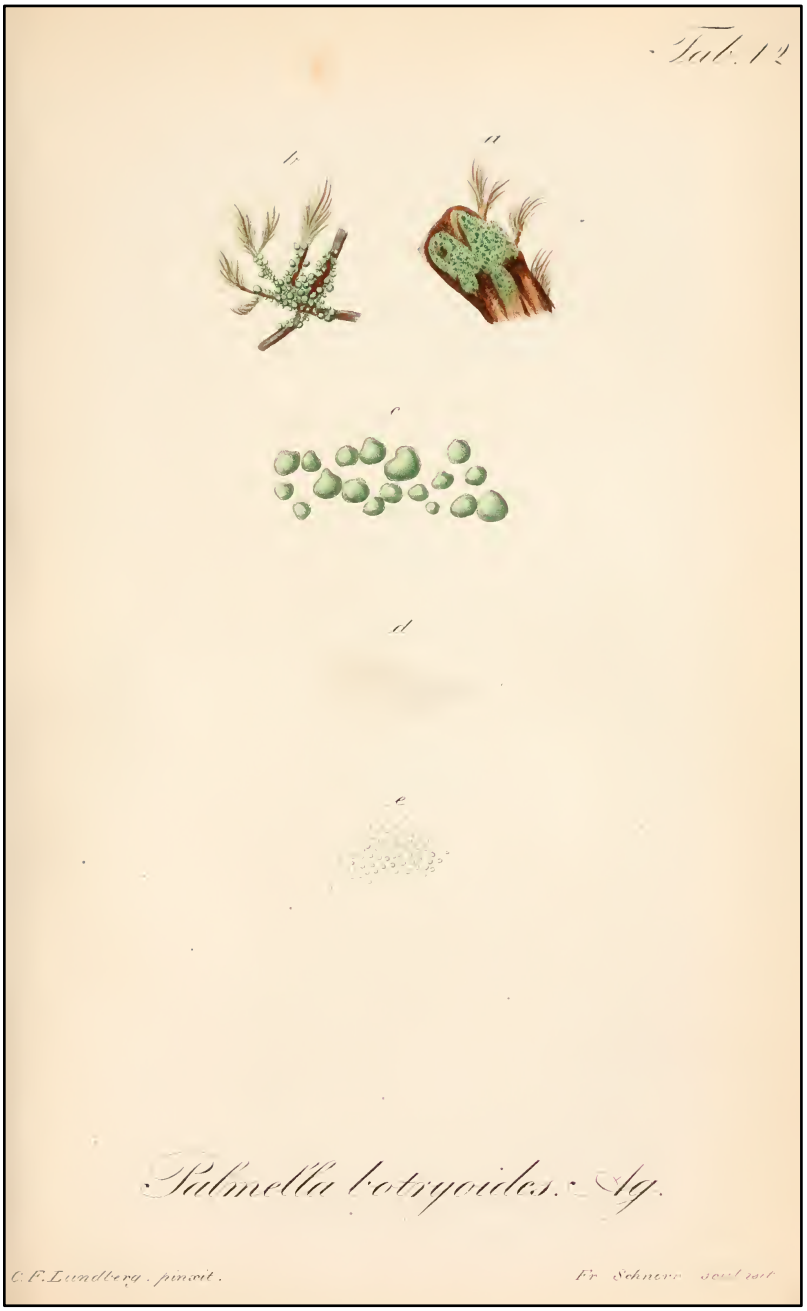


Figure 5. Possible early drawing of *Botryococcus braunii*. This illustration was drawn by the Swedish botanist Carl Agardh in 1835. The globules approximately resemble the colony shapes commonly observed in species of *Botryococcus*. However, the lack of detail makes the illustration difficult to interpret. Figure reprinted from Agardh, 1835 *Icones algarum europaearum: représentation d'algues européennes suivie de celle d'espèces exotiques les plus remarquables récemment découvertes.* L. Voss.

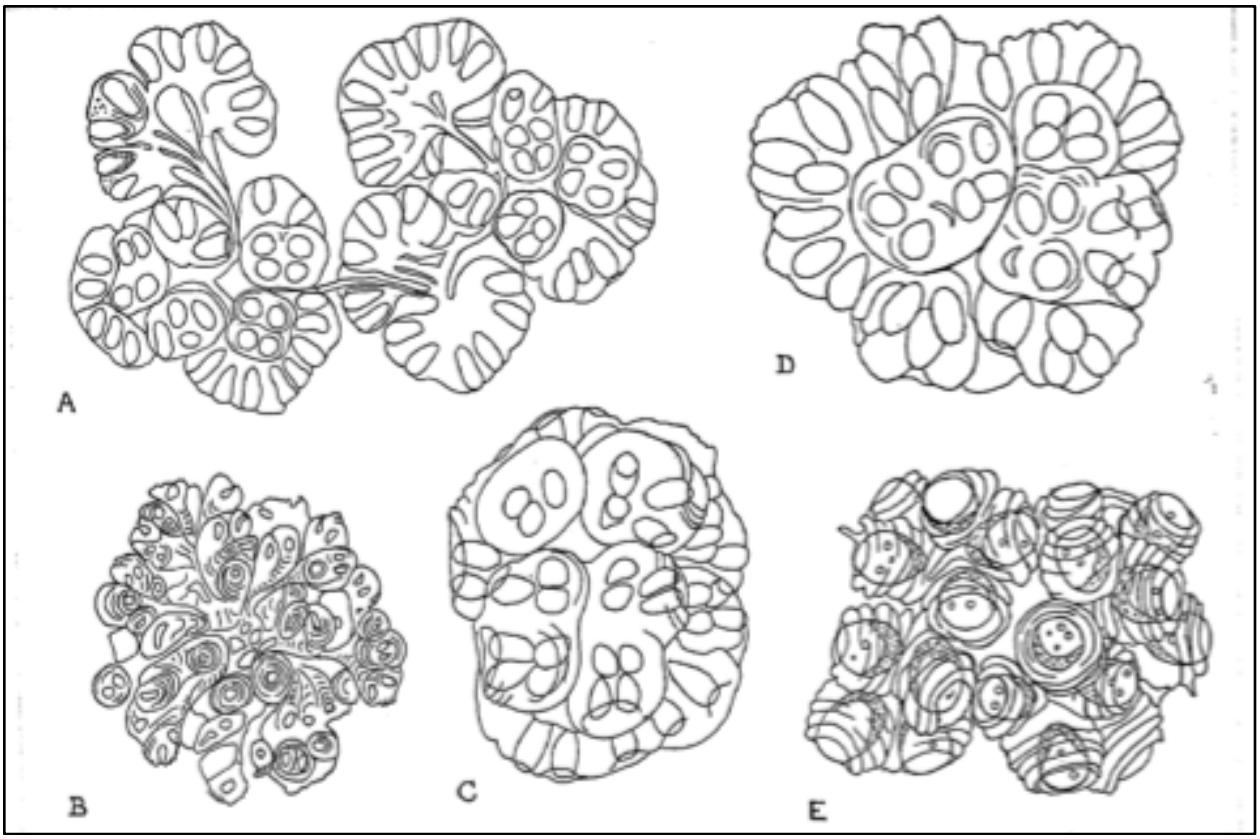


Figure 6. Detailed drawings of *Botryococcus braunii* colony morphology. This figure shows the remarkably accurate drawings of *B. braunii* colonies by Kathleen Blackburn in 1936. These drawings were made following careful observations under a light microscope in combination with various dye stainings. Figure reprinted from Blackburn, 1936, *Trans Roy Soc Edin* **58:**841-854.

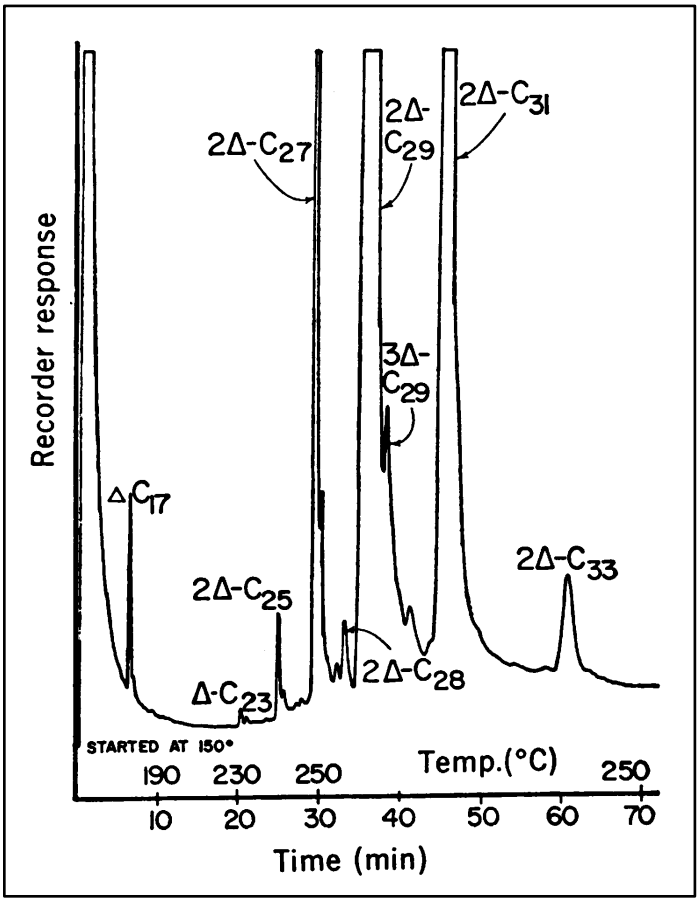


Figure 7. Gas chromatographic separation of hydrocarbons from *Botryococcus braunii*. This figure shows the first ever application of gas chromatography to analyze oils from *B. braunii*. The instrument was an F&M 800 gas chromatograph equipped with a flame ionization detector. The glass column, 1.7 m by 0.3 cm inside diameter was packed with OV-1 (methyl silicone fluid). Figure reprinted from Gelpi et al., 1968, *Science* **161:**700-701.

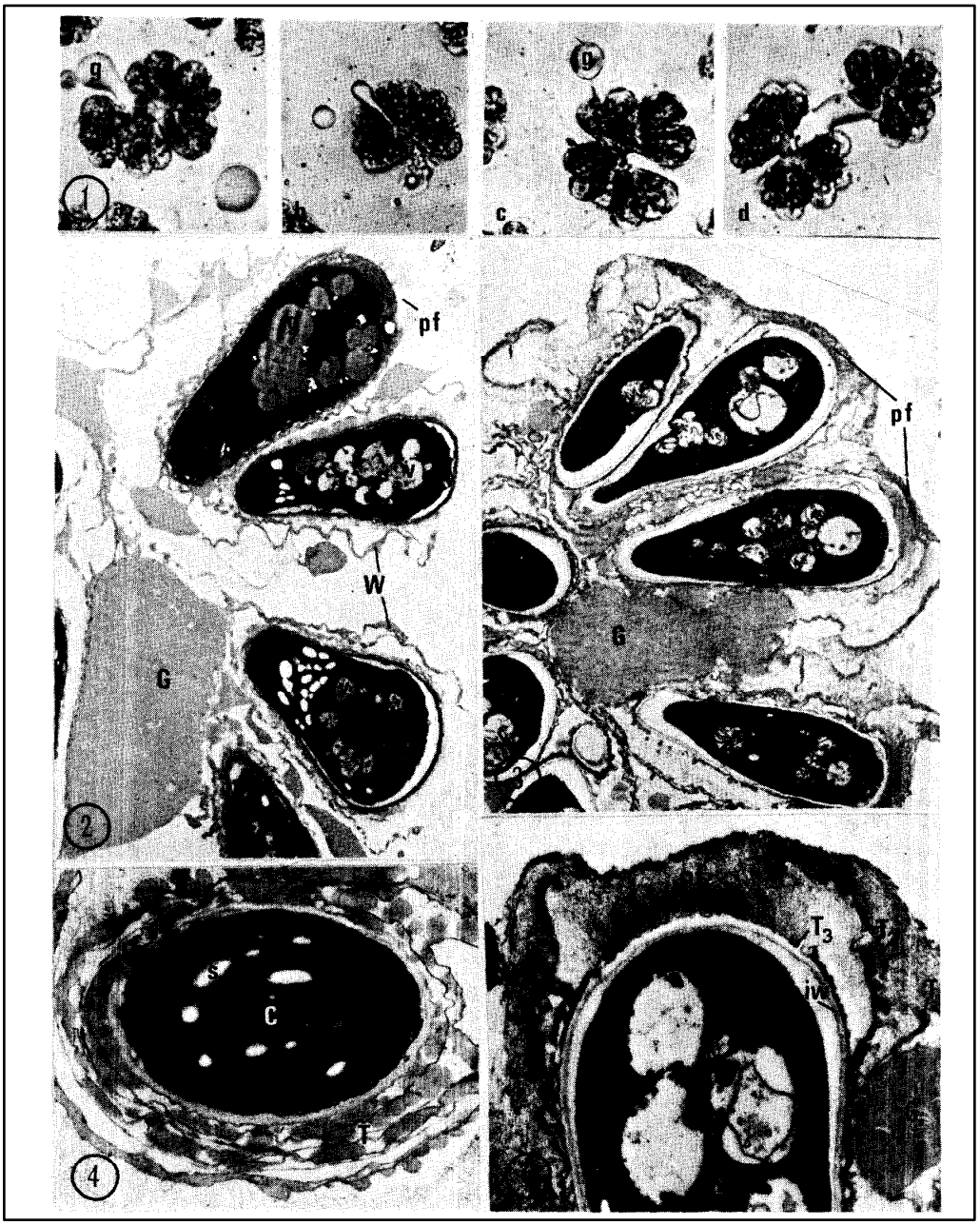


Figure 8. Light micrographs and electron micrographs of *Botryococcus braunii*. This figure shows some of the earliest published images of *B. braunii* taken with an electron microscope. These data mark an important step forward in the qualitative analysis of *B. braunii* cellular and colonial morphology. Figure reprinted from Largeau et al., 1980, *Phytochemistry* **19:**1043-1051.

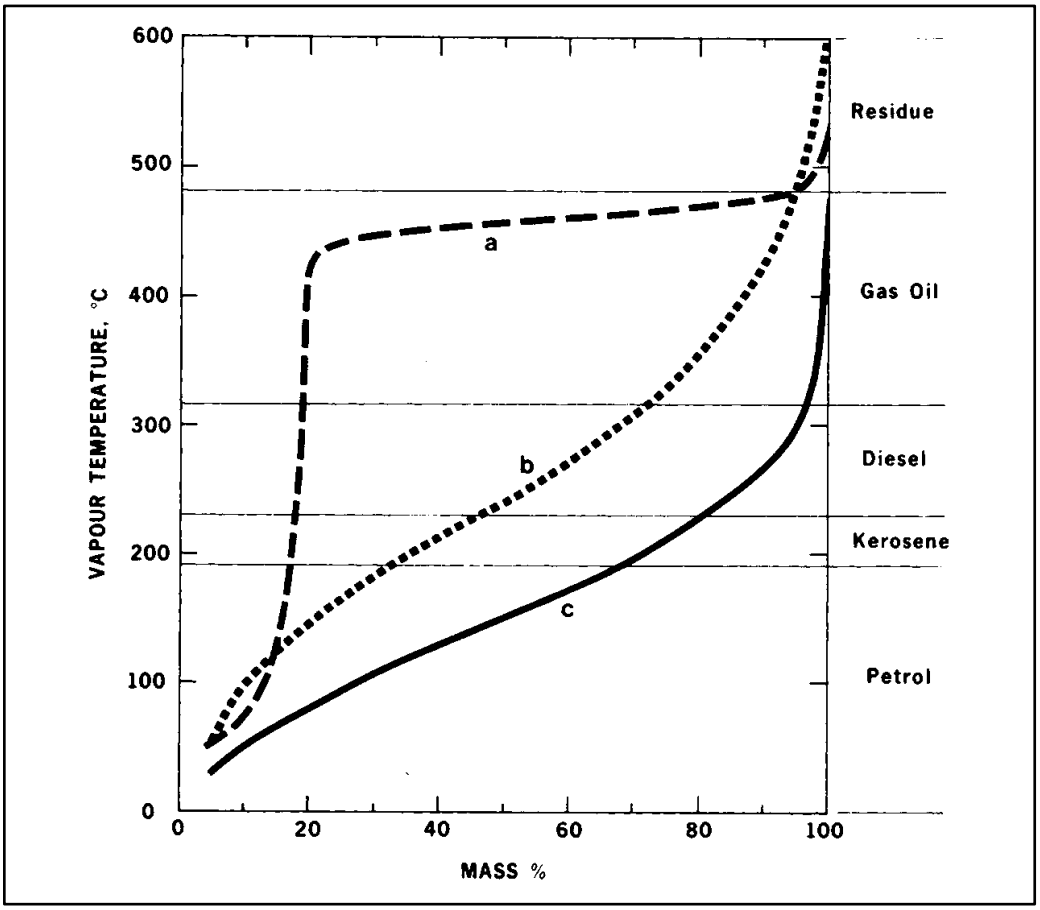


Figure 9. Comparison of boiling point ranges. This figure shows the yield curve for (a) unprocessed *Botryococcus* oil, (b) hydrocracked *Botryococcus* oil, and (c) typical Bass Strait crude oil. The data demonstrate that hydrocracked *Botryococcus* oils have a similar yield curve to a standard crude oil. Figure reprinted from Hillen et al., 1982, *Biotechnol Bioeng* **24:**193-205.

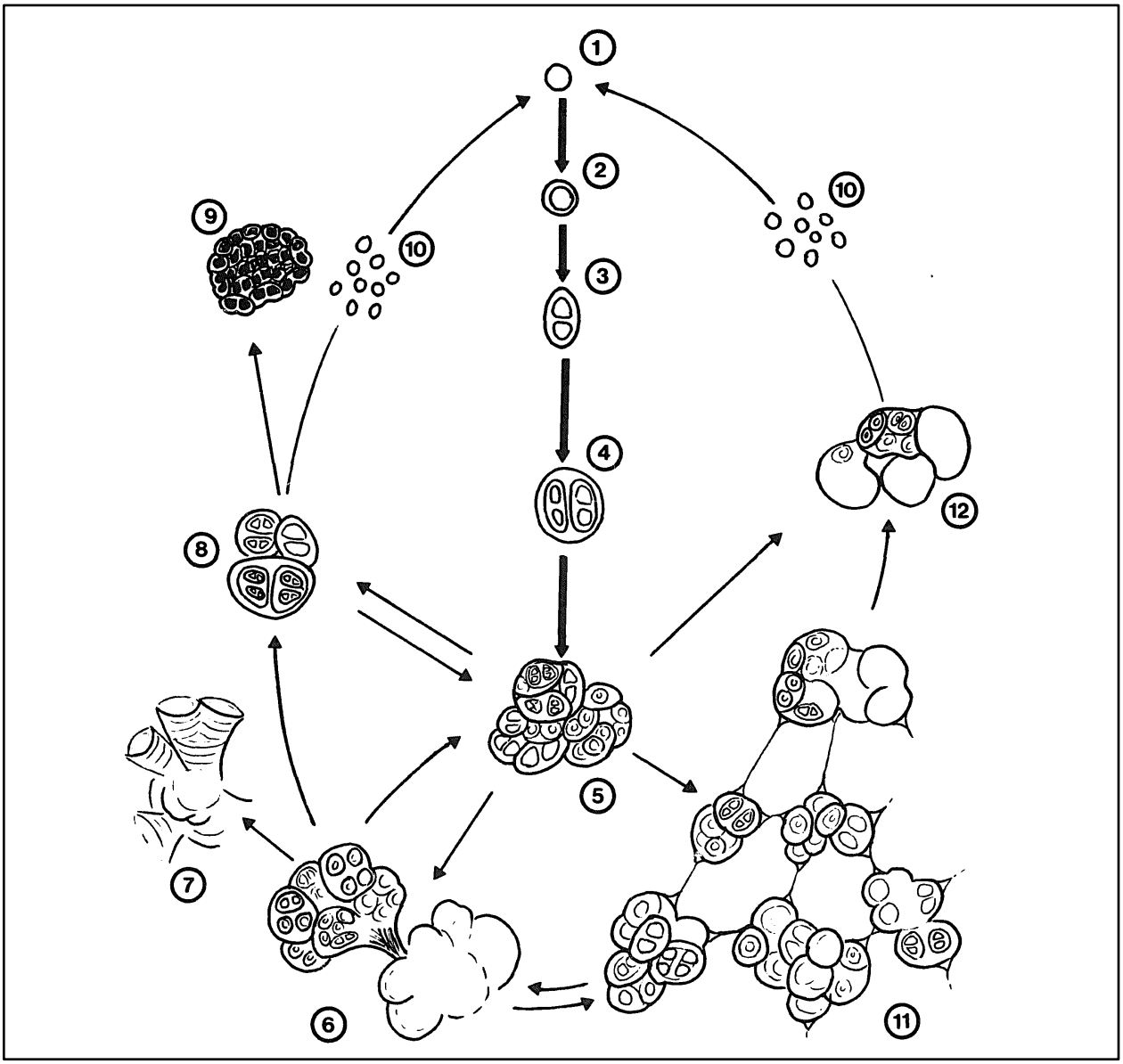


Figure 10. Description of the main developmental stages in the *Botryococcus braunii* life cycle. This figure shows (1) single autospore; (2) single autospore with first cup secreted; (3) first longitudinal division of the autospore; (4) second division, longitudinal but perpendicular to the first; (5) simple unbranched compound colony; (6) branched compound colony; (7) old matrix with "growth" rings and colonies already detached by fragmentation; (8) simple compound colony obtained by fragmentation; (9) skeleton matrix with empty cups; (10) dispersed autospores; (11) large complex of compound colonies held together by mucilaginous strands; (12) simple compound colony. Figure reprinted from Guy-Ohlson, 1992, *Review of Palaeobotany and Palynology* **71:**1-15.

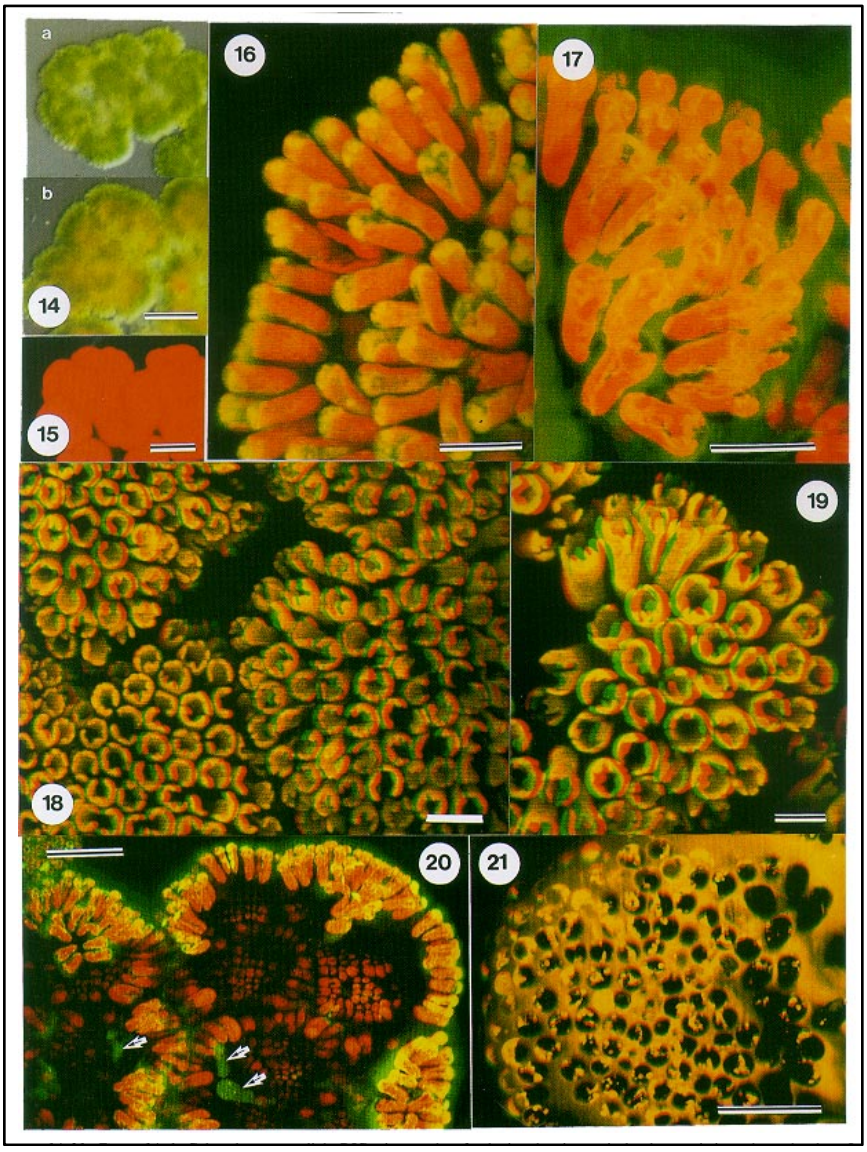


Figure 11. Light micrographs and confocal fluorescence micrographs of Botryococcus braunii. This figure shows: (14) light DIC micrographs demonstrating natural color variations, scale bar = 100 µm; (15) fluorescence DIC micrograph; (16) superimposed z-series projections of a stained colony showing plastid autofluorescence (red) and lipophilic material (yellow/green), scale bar = 20 µm; (17) z-series projections of a stained colony showing how the reticulate system sits outside the plastid and arches over the cell apex, scale bar = 20 µm; (18) stereo projection of the z-series showing chlorophyll autofluorescence from a cluster of colony margins, scale bar = 30 µm; (19) stereo projection of the z-series showing chlorophyll autofluorescence from a cluster of cells, scale bar = 10 µm; (20) superimposed z-series projections of a stained colony showing lipophilic material and secreted lipid droplets (arrows) resolved from the plastids, scale bar = 50 µm; (21) stereo projection of the z-series showing a stained colony that reveals lipophilic extracellular matrix surround cells, scale bar = 50 µm. Figure reprinted from Beakes and Cleary, 1998, *Journal of Applied Phycology* **10**:435-446.

1.2.2 The 21st Century

The beginning of the new millennium saw the first attempt at determining the biosynthetic mechanism of botryococcene. In early 2000, Okada *et al* from Japan and Kentucky published a paper on the cloning and characterization of squalene synthase from *B. braunii* (*110*). Because of the structural similarity between squalene and botryococcene, the authors hypothesized that this enzyme may have evolved promiscuous activity in *B. braunii* and become responsible for botryococcene biosynthesis. Although expression of this gene in *E. coli* did not result in the production of botryococcene, their work would continue and eventually yield important achievements. Meanwhile, in 2002, the first substantial review of *B. braunii* was published since the review by Wolf nearly two decades prior (*111*). Although the authors, Banerjee *et al* from India and New Zealand, had not previously published any work of their own on *B. braunii*, they succeeded where Guy-Ohlson had failed ten years before. Their review comprehensively summarizes the major achievements in the 1980s and 1990s by the CNRS group and others, and provides an outstanding overview of the diverse knowledge in the field of *B. braunii*. Although the biosynthetic mechanism for botryococcene was still unknown, in 2003, Sato *et al* from Japan (including Okada as corresponding author) conducted an experiment to determine the pathway used for biosynthesis of isoprenoid precursors in *B. braunii* (*112*). The two universal precursors for isoprenoid biosynthesis, isopentenyl diphosphate (IPP) and dimethylallyl diphosphate (DMAPP), are produced either by the mevalonate (MVA) pathway or the methylerythritol-4-phosphate (MEP) pathway. Both of these pathways are present in plants and are typically compartmentalized, with the MVA pathway active in the cytosol and the MEP pathway active in the chloroplast. By feeding ^{13}C-labeled glucose to *B. braunii* and tracing the isotope by nuclear magnetic resonance spectroscopy, Sato *et al* concluded that the MEP pathway is principally

responsible for providing the building blocks of botryococcene. The following year, Okada *et al* published the development of an *in vitro* enzymatic assay for the detection of botryococcene synthase activity (*113*). While such an assay is a vital tool for the characterization of an enzymatic mechanism, it would still be a few more years until they finally solved the puzzle of botryococcene biosynthesis.

Nearly ten years after the last phylogenetic analysis, in 2004, Senousy *et al* from the University of Newcastle upon Tyne conducted a new phylogenetic placement of *B. braunii* (*114*). Again using the sequences of nuclear small subunit ribosomal RNA genes, they constructed phylogenetic trees using several statistical methods. The accumulation of new data over the last decade enabled them to draw upon 53 previously reported rRNA genes from other species in the Chlorophyta. With this improvement in taxon sampling, they confidently concluded that *B. braunii* is a member of the Trebouxiophyceae, still the current view, and its closest relatives with available sequence data are in the genus *Choricystis*. In 2005, the leading experts of the time on *B. braunii*, Metzger and Largeau, finally published a review on the species (*115*). In it they focused on the array of complex natural products that had been isolated from the alga and chemically characterized. Significantly, they proposed a schematic of the biosynthetic pathway for botryococcene, supported by previous studies, which accurately depicted the mechanism that would eventually be discovered. However, they did not discuss possible pathways for the biosynthesis of lycopadiene, due to a complete lack of knowledge about this process, although this would also eventually be discovered. While they did discuss possible pathways for the biosynthesis of alkadienes, supported by some of their early work, to this day the exact mechanism is still unknown. Three years later, in 2008, the final work by members of the CNRS group was published (*116, 117*). While their work had slowed down substantially in the mid-nineties, their final

publications marked the end of an era. The achievements of this group cannot be overstated, as they transformed the field of *B. braunii* research, setting the stage for more advanced analyses, and inspiring subsequent generations of researchers around the world to work on this interesting organism.

Beginning in 2010, the field of *B. braunii* research started to change and grow rapidly. That year, Weiss *et al* from Texas A&M University (in collaboration with scientists from Japan and Kentucky) published the first ever estimation of the *B. braunii* race B (Showa strain) genome size, at 166.2 ± 2.2 Mb (*118*). The genome size was estimated with flow cytometry analysis of intact nuclei isolated from the alga and using *Drosophila virilis* nuclei as a standard. The following year, Weiss *et al* used this technique again to estimate the genome sizes of *B. braunii* race A (Yamanaka strain) at 166.0 ± 0.4 Mb, and race L (Songkla Nakarin strain) at 211.3 ± 1.7 Mb (*119*). While these estimates are tremendously useful and served as a starting point for further genomics analyses of the strains, the insights are limited by the small number of strains subjected to analysis. There may be substantial genetic variation across ecotypes, which would not be apparent from surveying only three strains. Beyond steps in the direction of genomics, the year 2011 also heralded another major advancement in the field of *B. braunii* research: discovery of the biosynthetic mechanism for botryococcene. Niehaus *et al* from the University of Kentucky (in collaboration with scientists from Japan and Texas) found three squalene synthase-like (SSL) genes in *B. braunii*, which work together in a hitherto unseen mechanism (*120*). In summary, SSL1 catalyzes the biosynthesis of presqualene diphosphate (PSPP) from two molecules of farnesyl diphosphate (FPP), releasing one molecule of pyrophosphate. The PSPP intermediate is then converted either to squalene by SSL2 or botryococcene by SSL3, with both of them requiring NADPH. This result was highly unexpected, as it was assumed that a single enzyme and not two enzymes would

conduct the whole biosynthetic process of botryococcene. Following up on this work, in early 2012, Niehaus *et al* identified some of the methyltransferase enzymes responsible for methylation of botryococcene (*121*). Quite significantly, this study was the first in the field of *B. braunii* to make use of a transcriptome for gene identification, followed by cloning and characterization, and clearly demonstrated the power of this approach. Later that same year, Molnar *et al* from the University of Arizona (in collaboration with scientists from Texas) published a comprehensive transcriptomic analysis of *B. braunii* race B metabolism (*122*). A custom pipeline was used to assemble 46,422 transcripts from 1.3 million pyrosequencing reads obtained with a Roche 454 GS FLX Titanium DNA sequencer. Functional annotations were assigned to 20,906 of the transcripts and manual curation enabled the group to reconstruct numerous metabolic pathways of interest. Although it was not the first transcriptome reported for *B. braunii*, it did provide an immensely valuable resource for subsequent studies.

The first transcriptome analysis of *B. braunii* was also published in 2012, but a few months before Molnar *et al*. This report, from Baba *et al* in Japan (not associated with Okada *et al*), focused on a strain of *B. braunii* race A (*123*). Utilizing pyrosequencing technology as described above, they obtained 185,936 reads and assembled them into 29,038 non-redundant transcipts, of which 964 were functionally annotated. They queried the transcriptome to identify various genes related to fatty acid biosynthesis, but were unable to draw any substantial conclusions from the data. Similarly, they published an almost identical analysis of the transcriptome for a race B strain at the same time (*124*). The race B dataset consisted of 209,429 pyrosequencing reads assembled into 27,427 non-redundant transcripts, of which 725 were functionally annotated. Again, they identified a few genes of interest for isoprenoid metabolism, but did not reach any substantial conclusions. The fraction of high-quality transcripts in these transcriptome assemblies was very low, as clearly

evidenced by the weak assignment of functional annotations. Nonetheless, in an effort to gain deeper insights into the metabolic differences between races A and B, they conducted a comparative transcriptomic analysis between their two datasets (*125*). Briefly, they found that genes encoding acyl-ACP elongation, acetyl-CoA carboxylase, 3-oxoacyl-ACP synthase, acyl-ACP desaturase, and stearoyl-CoA 9-desaturase enzymes were more highly expressed in race A. Conversely, they found that genes encoding geranyltransferase and squalene synthase enzymes were more highly expressed in race B. Finally, they conclude that in race A fatty acid elongation proceeds at first in an ACP-bound form and then a CoA-bound form; and in race B the pentose phosphate pathway feeds products directly to the MEP pathway. However, due to the poor quality of the transcriptome data and the lack of a robust experimental design, their conclusions are somewhat suspect. Astonishingly, this group published a fourth paper at the same time, in the same journal as the last three. This publication focused on the relationship between hydrocarbons and the phylogeny of *B. braunii* strains (*126*). Using 31 axenic strains isolated from various locations in Japan, they amplified and sequenced the 18S ribosomal RNA genes and constructed a phylogenetic tree. In parallel, they characterized the hydrocarbon content of each strain by gas chromatography and mass spectrometry. Incredibly, they discovered a new race of the species, which produces epoxy-n-alkanes and saturated n-alkanes, and have termed this race S. The phylogenetic data showed the clear segregation of strains according to race, demonstrating that the hydrocarbon profile is a strong proxy for phylogenetic placement. While they proposed dividing *B. braunii* into at least two different species, this has not yet gained much traction with the wider community of scientists working on *B. braunii*, and no formal changes have been made to the taxonomy.

At the end of 2012, Weiss *et al* (in collaboration with scientists from Missouri and Georgia) released another paper, focused on the colony morphology of *B. braunii* race B (*127*). Making use of a quick-freeze deep-etch sample preparation method for electron microscopy, they obtained new levels of ultrastructural detail. Furthermore, they achieved insights into the monomers comprising the polysaccharide sheaths that encapsulate cells and found a single, unknown protein that is associated with them. Although they did observe contact between lipid bodies, the chloroplast, and the endoplasmic reticulum, they did not see evidence of lipid body secretion. They argue instead that hydrocarbons are produced by the endoplasmic reticulum and delivered directly to the cell membrane, where they pass through the cell wall and enter the extracellular matrix. In direct conflict with this conclusion, in 2013, Suzuki *et al* from Japan (associated with Okada *et al*) published another ultrastructural analysis, where they argued for lipid body secretion (*128*). Using fluorescence and electron microscopy, they studied changes in lipid body structure and abundance through the cell cycle of growth and division. They observed that lipid bodies increase in number and size just as cells begin to divide and then disappear just after the formation of daughter cells. They hypothesize that lipid bodies produced during the growth of *B. braunii* are directly related to the accumulation of extracellular hydrocarbons through a secretion mechanism. Prior to their analysis of race B, that same year the group had published a similar analysis of race A (*129*). They determined the stage of hydrocarbon synthesis during the cell cycle by synchronizing cultures with aminouracil, feeding ^{14}C-labeled acetate, and measuring label incorporation with autoradiography. They additionally made use of light, fluorescence, and electron microscopy to study ultrastructural changes during the cell cycle. They observed the formation of numerous sites of contact between the chloroplast, mitochondria, lipid bodies, and the endoplasmic reticulum, suggesting the direct exchange of metabolites between these organelles. They found that hydrocarbon biosynthesis was

at its maximum just after septum formation between dividing daughter cells. They concluded that hydrocarbon precursors are synthesized in the cytoplasm, secreted to the cell surface, and then converted to the final hydrocarbon products. This is in contrast to the conclusions for race B, which appears to directly secrete mature hydrocarbons into the extracellular matrix. From all of these studies, it is clear that many questions remain about the mechanisms and sites of hydrocarbon biosynthesis, storage, transport, and secretion. Resolving these outstanding issues will require not only more detailed studies with existing experimental tools and approaches, but also the development of new tools and approaches. One such tool that could be tremendously useful is microfluidic technology. In 2014, Kim *et al* from Texas A&M University reported the first ever creation of a microfluidic device to capture, cultivate, and monitor *B. braunii* colonies (*130*). With the flexibility of microfluidic devices, it seems that they are only limited by the imagination of the researchers designing them. Undoubtedly they will be important in the years to come.

Almost a decade after the last review of *B. braunii*, in 2014 one was published by John Volkman, a geochemist from Australia (*131*). He brings a unique perspective to the field of *B. braunii* by elegantly weaving together themes from geochemistry and molecular biology. In particular, he emphasizes how biomarkers have proven useful in geochemical studies, which in turn provide a potential timeline for the evolution of specific biosynthetic pathways. Applying this logic to *B. braunii*, he estimates that the contemporary biosynthesis pathways for botryococcene and lycopadiene evolved no more than 55 Ma, in the early Eocene. Since fossil remains of *B. braunii* are clearly and confidently identified in sediments much older than this, he argues that the ancestral species more closely resembled races A or S, producing non-isoprenoid alkyl chains. Developing a deeper holistic understanding of the evolutionary history of *B. braunii* and the biosynthetic pathways it possesses requires further advancements in the knowledge of its

molecular biology. This endeavor would be enormously assisted if researchers had the ability to genetically manipulate this species. In 2015, Berrios *et al* from Chile reported a method for the genetic transformation of *B. braunii* (race A) for the first time (*132*). To achieve transformation, they first weakened the cell wall by treatment with cellulase enzyme, and then subjected the treated colonies to electroporation in the presence of their DNA vector. The plasmid they used was pSI103, which contains the gene *AphVIII* as a selective marker, giving resistance to paromomycin. They confirmed the transformation by polymerase chain reaction and Western blot analyses of transformants and controls, and conclude it is a stable transformation. While this work is a very exciting advancement for the field of *B. braunii*, no other researchers have yet independently reproduced it. Moreover, it is of significant concern how the usage of antibiotics might affect the cultivation of *B. braunii*, which is well known to depend on symbiotic interactions with certain species of bacteria. For example, in 2015, Tanabe *et al* from the University of Tsukuba in Japan reported that an alphaproteobacterial endosymbiont substantially enhances the growth rate and hydrocarbon production of *B. braunii* (*133*). Furthermore, in 2016, Jones *et al* from the University of Exeter in England isolated a number of bacteria from a laboratory culture of *B. braunii* and identified them by whole genome sequencing (*134*). The removal of bacteria from cultures of *B. braunii* could have the unintended effect of impeding growth and hydrocarbon production, or otherwise negatively affecting its fitness. While this may be acceptable for basic scientific inquiries into the molecular functions of the species, it is certainly important to consider such consequences for potential commercial applications.

As 2016 progressed, researchers continued to publish important achievements in the study of basic *B. braunii* physiology. Thapa *et al* from Texas A&M University published the long-unknown biosynthetic mechanism of lycapodiene in race L (*135*). Taking a similar approach to

that used in the identification of the botryococcene biosynthesic mechanism, they began by searching for squalene synthase-like genes in the transcriptome of *B. braunii* race L. Just two genes were identified, and subsequently cloned and tested for enzymatic activity. They found that one gene was a true squalene synthase, but the other was found to be responsible for the biosynthesis of lycopaoctaene, and thus termed lycopaoctaene synthase (LOS). While the LOS enzyme primarily accepts two molecules of geranylgeranyl diphosphate to form lycopaoctaene, they found that it has significant promiscuous activity, also accepting farnesyl diphosphate and phytyl diphosphate as substrates. Nonetheless, the evidence indicates that LOS first forms lycopaoctaene and then still-unknown reductase enzymes reduce all but two of the double bonds, yielding lycopadiene. In the domain of genomics, Blifernez-Klassen *et al* from Bielefeld University in Germany published the complete chloroplast and mitochondrial genome sequences for the Showa strain of *B. braunii* race B (*136*). The chloroplast genome is 156,498 bp, with GC content of 41.51%, and 105 putative protein-coding genes, 31 tRNA genes, and 3 rRNA genes. The mitochondrial genome is 129,356 bp, with GC content of 50.41%, and 43 putative protein-coding genes, 23 tRNA genes, and 3 rRNA genes. These sequences are very useful for phylogenomic analyses as well as assessments of diversity and evolution among strains of *B. braunii*. To this end, it would be useful to obtain more organellar and nuclear genome sequence data for the strains of *B. braunii* that have been collected and cultured in the laboratory. In 2017, Browne *et al* from Texas A&M University (in collaboration with scientists from Alabama, Arizona, California, Kentucky, New Mexico, and Japan) reported the first nuclear genome sequence of *B. braunii*, also for the Showa strain of race B (*137*). The final draft assembly consisted of 184,385,342 bp in 2,752 scaffolds (N50 = 373 kb) with 49.6% GC content and 1,148 gaps (4.611 Mbp). They predicted 18,726 genes with a mean of 5.7 exons per gene, a median exon length of 178 bp, and a median

intron length of 578 bp. They found that 1,437 scaffolds had no predicted genic content; accounting for 6,183,350 bp, though all these scaffolds were fairly small and the largest was 49,840 bp. Also in 2017, Sambles *et al* from the University of Exeter in England published a metagenomic analysis of microbes associated with cultures of *B. braunii* race B (strain Guadaloupe), tracking changes in the consortia after perturbation with antibiotics (*138*). They found that the species most strongly associated with the alga included members of the Rhizobiales, such as the genera *Bradyrhizobium* and *Methylobacterium*, as well as members of the genera *Dyadobacter*, *Achromobacter* and *Asticcacaulis*.

Another interesting development of 2017 was the publication by Deng *et al* from the Jiangsu University of Science and Technology in China, describing the first ever identification of microRNAs in *B. braunii* (*139*). Using an Illumina HiSeq 2000, they constructed and sequenced a small RNA library for *B. braunii*, obtaining 10 million reads. After processing this data, they were able to identify 42 known microRNA families and 14 novel microRNA families. Using gene ontology analysis, they determined that these microRNAs are putatively involved in the regulation of metabolic and cellular processes, gene expression, and stress/defense functions. This information is very important for determining specific regulatory mechanisms and unraveling the global regulatory landscape in *B. braunii*. However, there are many other molecular functions in *B. braunii* that benefitted from more detailed studies in 2017. Suzuki *et al* (associated with Okada *et al*) published an absolutely amazing 3-dimensional reconstruction of the endoplasmic reticulum and other intracellular structures (*140*). This work significantly advances the understanding of spatial organization in *B. braunii* cells and sheds new light on the processes of interaction between the chloroplast, oil bodies, endoplasmic reticulum, nucleus, and plasma membrane. The light harvesting complexes (LHCs) of *B. braunii* were purified and characterized for the first time by

van den Berg *et al* from Vrije Universiteit Amsterdam in the Netherlands (*141*). They found many similarities between the LHCII of *B. braunii* and that typical of higher plants, such as chlorophyll composition and pigment organization. In contrast, they found that *B. braunii* LHCII has loroxanthin instead of lutein, and higher content of red chlorophyll *a*, compared to higher plants, although this does not seem to affect excitation energy transfer or fluorescence lifetimes. This information is important for understanding the role of the photosynthetic machinery in governing the growth rate of *B. braunii*, which is quite slow compared to other species of Chlorophyta. Finally, the first *B. braunii* paper of 2018, by Tatli *et al* from Texas A&M University (in collaboration with scientists from Georgia) revealed the identity of the polysaccharide-associated protein (PSAP) found in the fibrillar cell caps (*142*). Studies like these continue to improve the understanding of basic *B. braunii* physiology. Such information is very useful for the development of *B. braunii* as a model organism.

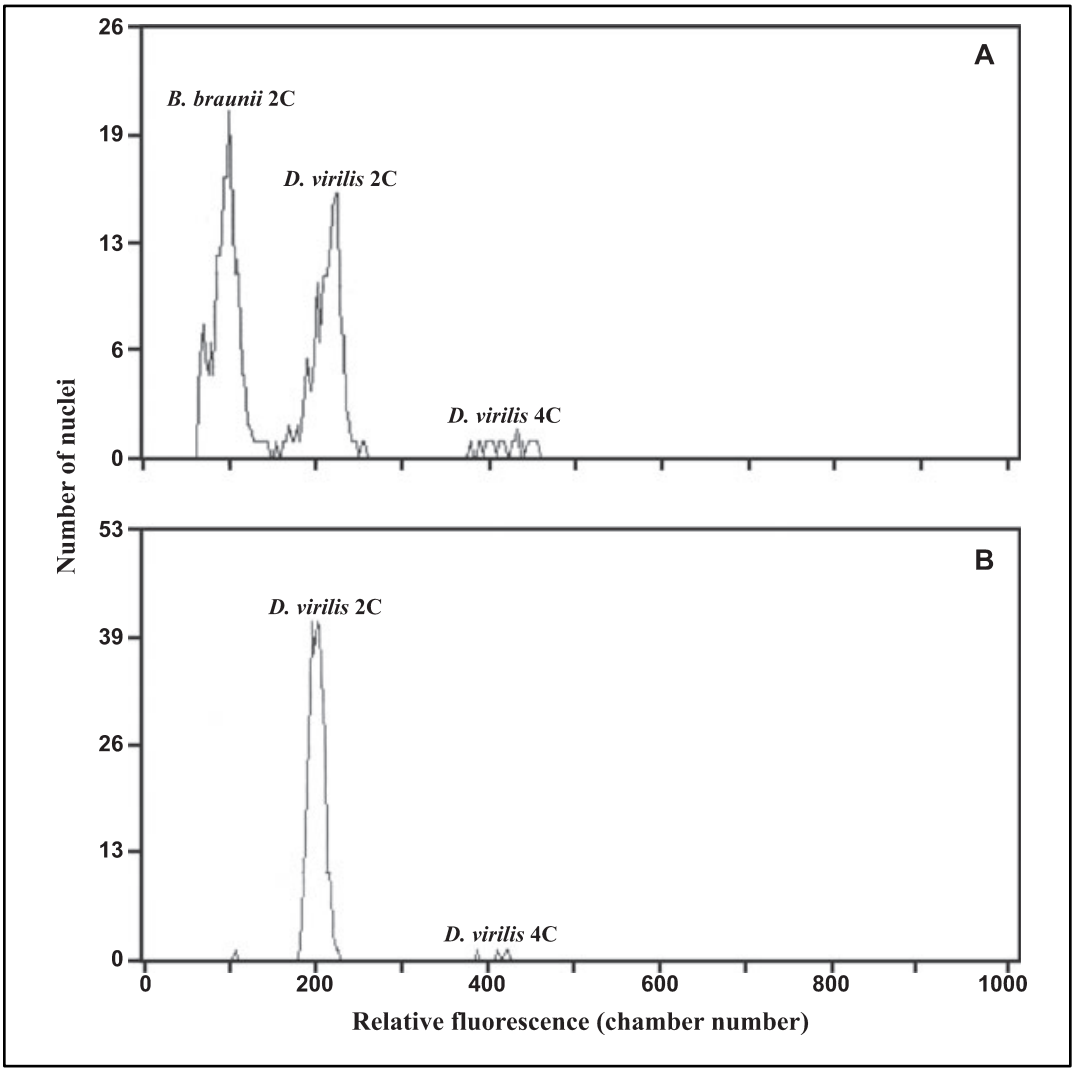


Figure 12. Flow cytometry analysis of Botryococcus braunii race B (Showa) for genome size determination. Diagrams show the number of nuclei with differing levels of red fluorescence from propidium iodide binding to DNA of (A) 2C nuclei of B. braunii, and 2C and and 4C nuclei of Drosophila virilis; and (B) 2C and 4C nuclei of D. virilis only. Based on these data, the B. braunii race B (Showa) genome size was estimated at 166.6 ± 2.2 Mbp. Figure reprinted from Weiss et al., 2010, *Journal of Phycology* **46**:534-540.

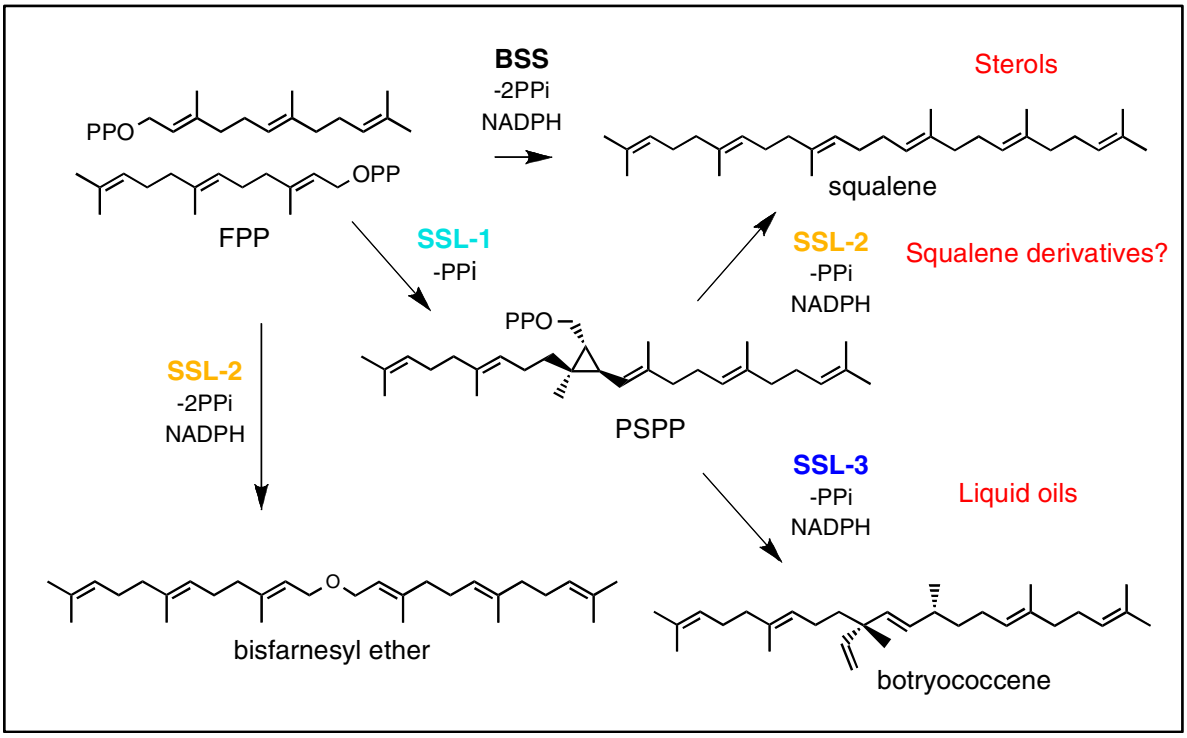


Figure 13. The catalytic roles of the squalene synthase-like enzymes in Botryococcus braunii race B. The previously identified squalene synthase gene (BSS) is thought yo provide squalene essential for sterol metabolism, whereas the squalene synthase-like genes SSL-1, SSL-2, and SSL-3 provide for the triterpene oils serving specialized functions for the algae. In combination with SSL-1, SSL-2 could provide squalene for extracellular matrix and methylated squalene derivatives, while SSL-1 plus SSL-3 generates botryococcene, which along with its methyl derivatives, accounts for the majority of the triterpene oil. Figure reprinted from Niehaus et al., 2011, *Proc Nat Acad Sci* **108**:12260-12265.

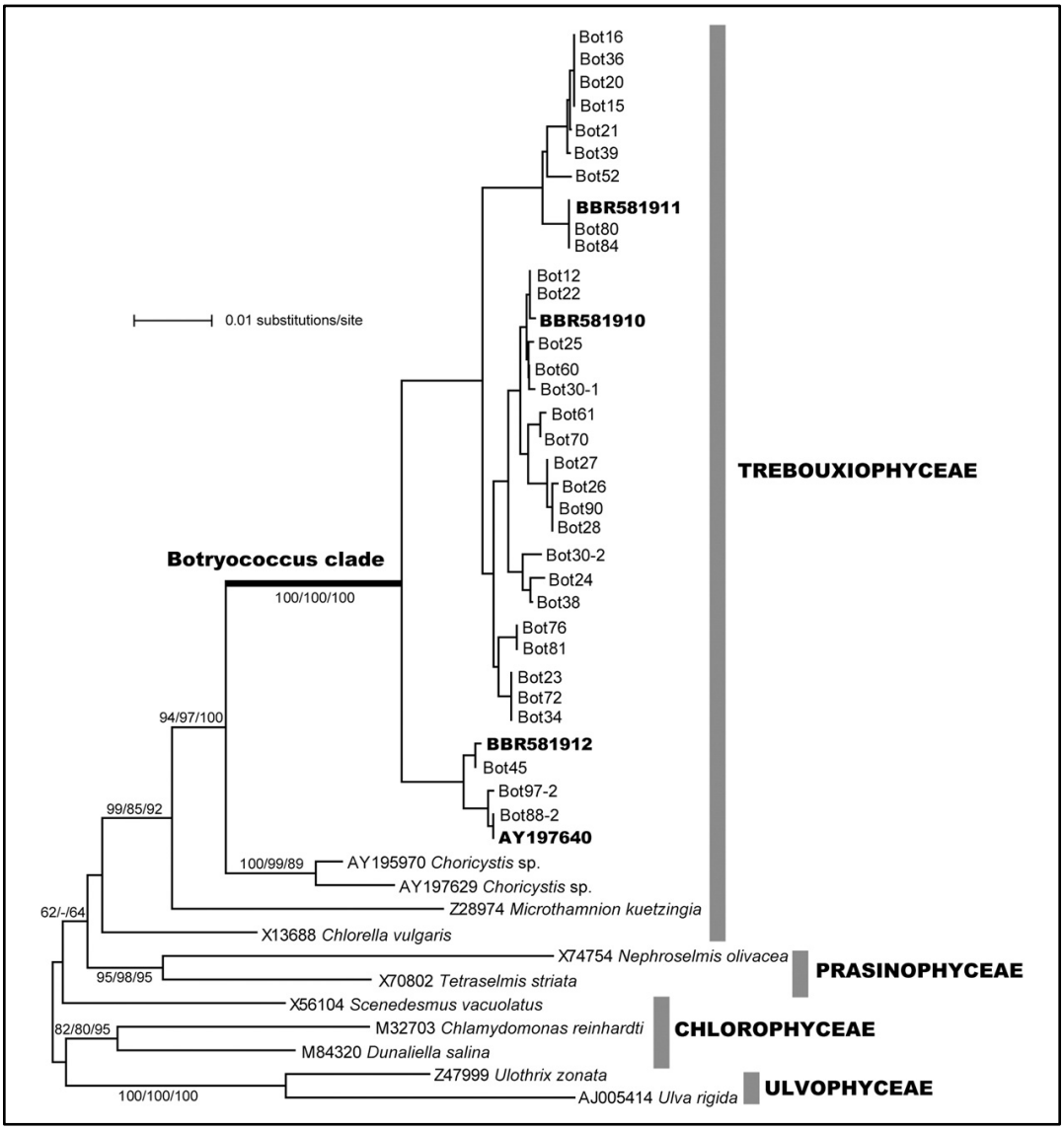


Figure 14. Phylogenetic tree of 18S rRNA gene sequences of Botryococcus and other Chlorophyte groups. The tree is rooted on the branch between the Prasinophyceae and the other chlorophytes. Numbers around the internodes indicate bootstrap values in the NJ, MP, and ML analyses (1000, 1000, and 100 replications, respectively). The bootstrap values in the Botryococcus clade corresponded to those in Fig. 2; not indicated in this tree. The accession numbers in the Botryococcus clade are the isolates with 18S rRNA sequences that were determined by Senousy et al. (2004). Figure reprinted from Kawachi et al., 2012, *Algal Research* **1**:114-119.

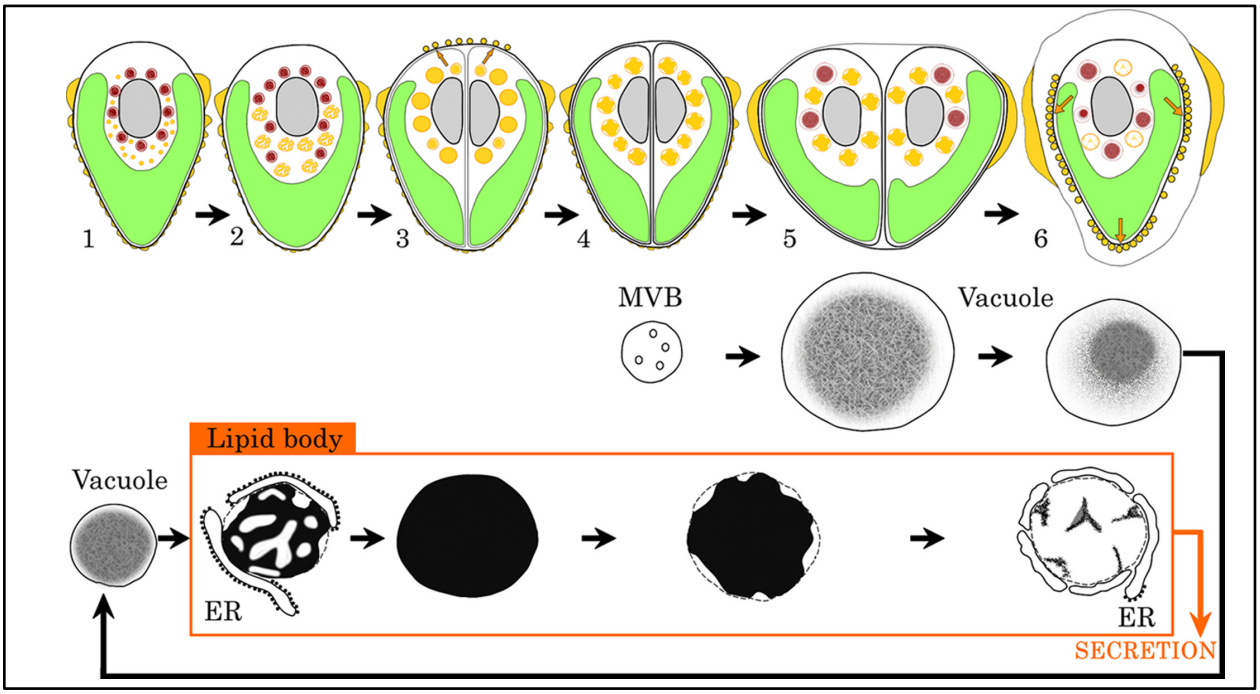


Figure 15. Transformation of lipid bodies and vacuoles during the cell cycle. The top line shows the growth stage of B. braunii. Yellow, lipid body in cytoplasm and lipid on the cell surface; red, vacuole; green, chloroplast; gray, nucleus; orange arrow, lipid secretion. The second and third lines show the transformation of lipid bodies and vacuoles. Figure reprinted from Hirose et al., 2013, *Eukaryotic Cell* **12**:1132-1141.

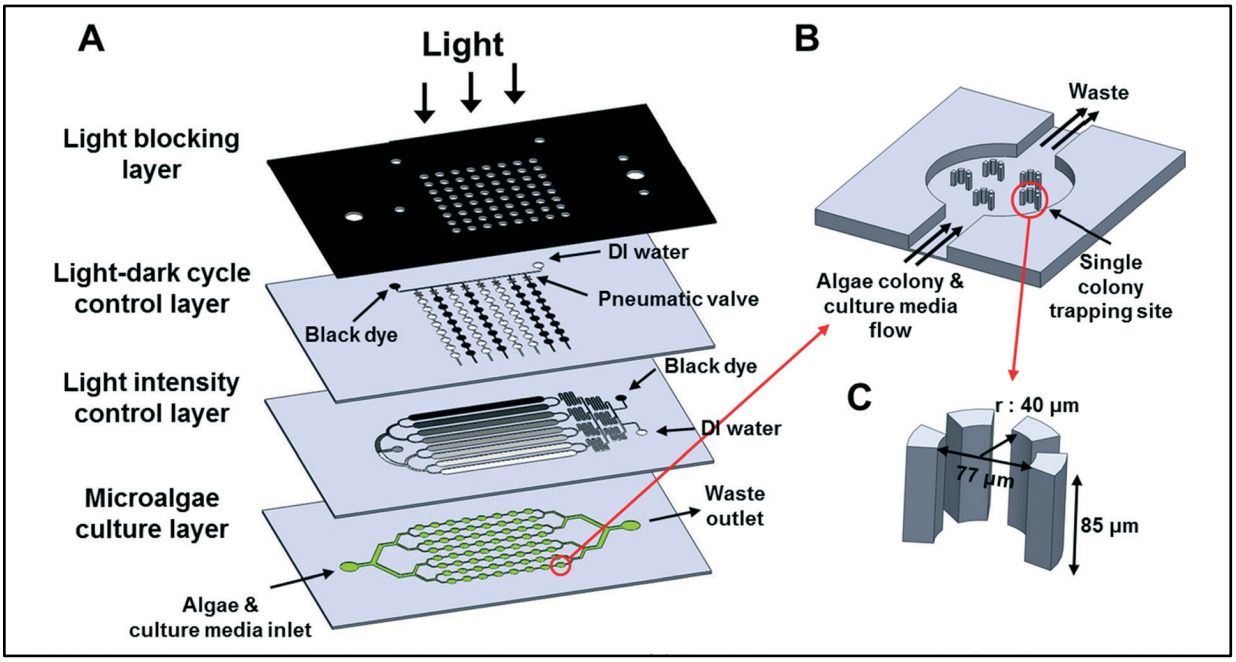


Figure 16. The high-throughput microfluidic microalgal photobioreactor array. (A) The platform was composed of four layers: a light blocking layer, a microfluidic light–dark cycle control layer, a microfluidic light intensity control layer, and a microalgae culture layer. (B) Enlarged view of a single culture compartment having five single-colony trapping sites. (C) A single-colony trapping site composed of four micropillars. Figure reprinted from Kim et al., 2014, *Lab on a Chip* **21:**47-58.

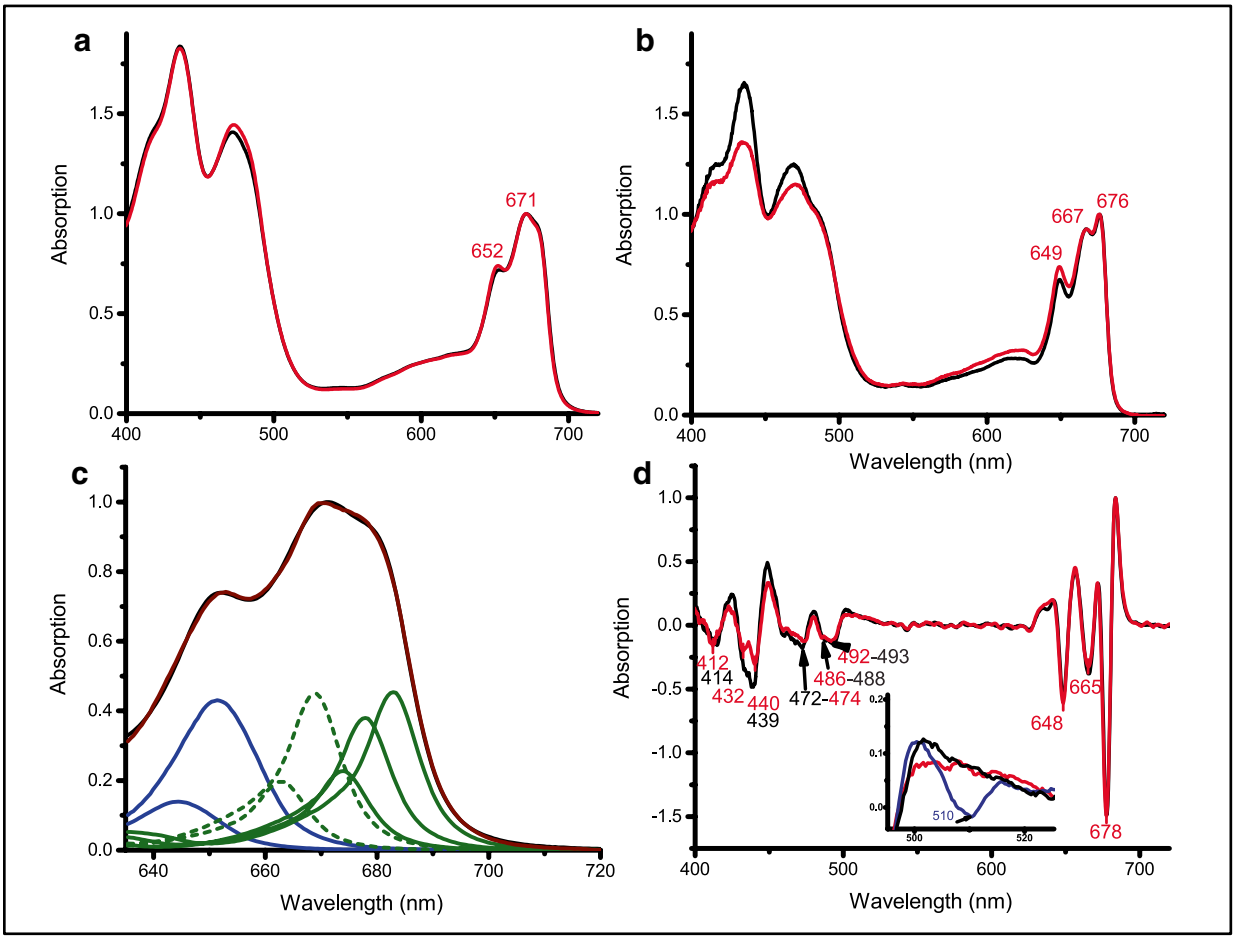


Figure 17. Absorption spectra, QY fitting and second derivative of the monomeric and trimeric fractions. (a) RT absorption spectra normalized to the QY maximum (monomers black, trimers red). (b) 77 K absorption spectra normalized to the QY maximum (monomers black, trimers red). (c) Absorption spectrum of trimeric complexes fitted with the spectra of Chl a and Chl b in protein environment (Cinque et al. 2000). Blue represents Chl b spectral forms, green represents Chl a spectral forms (solid: red spectral forms, dotted: blue spectral forms). The measured spectrum is in black and the fitting result in brown. (d) Second derivative spectra of the 77 K absorption spectra normalized to the 684 nm maximum. In a, b, and c labels indicate the same peak positions in both fractions. In d, black is the monomeric fraction, red to trimeric, and blue (inset) to AT LHCII trimers. Figure reprinted from van den Berg et al., 2017, *Photosynthesis Research* **27**:42-48.

1.3 Renewable Fuel and Synthetic Biology

While *B. braunii* is an interesting organism with a fairly long history of scientific study, it is only one piece in the broader picture of studies on algae biology and even broader biotechnology. The following sections detail some of the origins, motivations and recent advances in these two sectors, offering some insight into the direction of research yet to come.

1.3.1 Development of Algae Biofuel Technology

Since the oil crises of the early 1970s, there has been substantial interest in renewable fuel technology, as elegantly explained in a 1987 publication from Melvin Calvin, the godfather of photosynthesis and one of the most important biochemists of the 20th century (*143*). He laid out a vision for the production of fuel oils from plants including *Botryococcus braunii*, *Copaifera multijuga*, *Euphorbia lathyris*, and *Pittosporum resiniferum*. In the US at the time, seed oils from plants including peanuts, safflower, soybean, and sunflower, were already produced on a significant scale, and would soon form the basis of commercial scale biodiesel production (*143, 144*). In Brazil, sugar cane was grown on a massive scale and used to produce 7 billion liters of ethanol in 1985, approximately 20% of their liquid energy needs (*143*). By contrast, in 2001 the US was producing almost 6 billion liters of ethanol from cornstarch (*145*). However, the production of fuel ethanol from traditional food crops is not environmentally or economically efficient, due to the heavy requirement of herbicides, pesticides, fertilizers, irrigation, and machinery, which are required to grow the plants (*145*). Almost 30 years after Calvin published his vision of biofuels, in 2006, Ragauskas *et al* published a 21st century roadmap for biofuels (*146*). They discuss the concept of an integrated biorefinery for the combined production of biopower, biofuels, and biomaterials, to maximize the use efficiency of biomass. While many people share

enthusiasm for such projects, there are substantial challenges, barriers, and limitations to the utilization of natural photosynthetic processes for industrial-scale operations, as discussed in 2010 by Larkum (*147*). In particular, he highlights the inherent inefficiency of natural photosynthesis, dousing with cold water the hopes for large-scale, plant-based (especially algae-based) bioproducts. However, not everyone shares this pessimistic view, and many scientists view photosynthetic inefficiency as an opportunity for optimization. That same year, a very comprehensive review of photosynthetic efficiency was published by Zhu *et al* in which they discuss various bioengineering approaches to improving photosynthesis (*148*). They postulate that such efforts have the potential to more than double the yield of important industrial crops. Towards the end of 2010, Wijffels and Barbosa published an important and influential perspective on algal biofuels, outlining the state of the art and highlighting critical topics for further study (*149*). They estimated that an economical process for commercial scale cultivation of microalgae and conversion into diverse, valuable bioproducts could emerge in 10 to 15 years. Eight years later, and still no such platform has achieved widespread commercial success, but not for lack of effort. The last six years alone have seen an enormous amount of research on every aspect of producing biofuels from microalgae. Following is a brief selection, summary, and discussion of notable papers from this timeframe.

While there is great potential for algae, major roadblocks remain for commercial cultivation, as reviewed in 2013 by Chisti (*150*). Among these are an insufficient supply of concentrated carbon dioxide for cultivation, the ability to recycle phosphorous and nitrogen nutrients, a limited supply of freshwater, and a lack of processes for recovering energy from oil-extracted biomass. While he casts doubt on the near term viability of algae for production of fuel oil, he suggests that commercial production is still very possible in the long term, if specific

challenges are successfully met. Since his review, substantial progress has been made in addressing the issues he raised. Yoshida *et al* provided a well-rounded review in 2012 of the potential for algae to play a much more significant role in human civilization (*151*). Their high-level overview integrates biological, technical, and financial aspects of commercial-scale algal cultivation, and illuminates avenues for further research and development. As of 2013, numerous technologies were already available for harvesting algae and extracting oil, as reviewed by Pragya *et al* (*152*). The choice of methods for harvesting, extracting, and converting algal biomass will have a large impact on process economics and life cycle, as discussed below. Despite the diversity of available technologies, many of them have only been demonstrated at a laboratory-scale. This is a problem, as obstacles exist in translating laboratory-scale experiments to larger scales (*153*). Another important challenge facing the algae industry is the lack of standardized protocols for strain management (*154*). By contrast, traditional agriculture makes use of seed banks to tightly control cultivars and ensure crop quality. This lack of infrastructure also negatively impacts the utilization of transgenic strains, which suffer from inconsistency and gene silencing. Adopting better strain management practices should help to broadly improve algal cultivars and yield more consistent results.

Planning algal production systems is a complex process and requires the consideration of numerous environmental, economical, and technical factors. Miara *et al* developed the concept of the "energy-water-food nexus" (EWFN), a framework for assessing the sustainability and environmental impact of algal biofuel production systems (*155*). The EWFN framework is useful in the process of designing a production system and can help the planners choose the best technologies for maximizing sustainability and minimizing environmental impact. Due to the large variance of reported algae biomass productivity, it is challenging to gain insights into the potential

economic viability of this technology. One study attempted to tame the variability by developing a model that integrates a wide range of data for a specific cultivation platform for a single species (*156*). The authors integrated meteorological data from regions across the globe to determine the most feasible locations for algae cultivation and found that temperature is one of the most important factors affecting biomass yield. Another assessment of regional cost variability in biofuel production found that one of the major determinants of the variability is resource availability (*157*). While regional climate is a significant factor, other factors include the local availability of water, flue gas (i.e. carbon dioxide), and nutrients. The authors concluded that, using conventional open pond cultivation technologies, economical production of algae in the United States is mostly limited to sites in Texas and Florida. While open ponds have traditionally dominated the algal production industry, they have severe limitations in comparison with closed photobioreactors (*158*). For example, photobioreactors can support higher photosynthetic efficiency, biomass concentration and productivity, enable superior control of conditions, and mitigate contaminating organisms. The challenge with photobioreactors is the lack of cost-effective designs, resulting in very high capital expenditures being required to build such systems.

Technoeconomic analyses are important for obtaining estimates of the financial viability of producing algal biomass. An economic modeling effort by Acien *et al* based on a validated manufacturing process showed that costs must be substantially reduced in order to be competitive (*159*). The process was based on the cultivation of *Scendesmus almeriensis* using tubular photobioreactors, followed by centrifugation and freeze-drying to obtain the final product. They estimated that with a production capacity of 200 tons per year, the production cost would be approximately $15 per kg, with labor and depreciation as the major factors contributing to this cost. A financial feasibility analysis by Richardson *et al* concluded that photobioreactors were less

economically competitive than open ponds for commercial scale cultivation of algae (*160*). They determined that in the base case, average total costs for lipid production were $12.73 per gallon and $31.61 per gallon for open ponds and photobioreactors, respectively. Based on their results they conclude that process innovations are needed in order to reduce production costs. Although automobiles may eventually shift to predominantly electrical power, certain forms of transportation, like aviation, will continue to be reliant upon liquid fuels. Technoeconomic analysis of microalgae cultivation for aviation fuel suggested that, with technologies in 2013, the minimum selling price would be $31.98 per gallon in the base case and about $8.33 per gallon in the best case (*161*). The authors found that facility costs accounted for 84% of the total capital investment, in particular the cost of harvesting equipment. Abodeely *et al* introduced in 2014 the Algae Logistics Model (ALM), which incorporated regional variation in climatic conditions and the corresponding impact on algae biomass productivity (*162*). Their technoeconomic analysis yielded a baseline production cost estimate of $16.83 per gallon of algal triglycerides for use in biodiesel production. Power plants burning coal and natural gas release enormous amounts of carbon dioxide into the atmosphere. Since algae require carbon dioxide for growth, it is possible that power plants could provide this important nutrient for industrial scale cultivation of algae. A comprehensive model of this process was developed and led to the conclusion that it could be profitable at a biodiesel price of $3.91 per gallon while also providing substantial reductions in GHG emissions (*163*). As previously mentioned, conversion technologies also have an impact on the final cost of the product. Bench-scale testing of hydrothermal liquefaction and subsequent modeling of process economics showed that this technology could be very useful for converting lipid-extracted algae biomass to liquid fuels (*164*). The authors estimated the minimum fuel-selling price between $2.07 and $7.11 per gallon, with feedstock, product yield, and equipment as the major factors affecting

cost. Silva *et al* developed and simulated a process for commercial scale production of biodiesel from algae, based on the best available technology and supporting data at the time (*165*). They concluded that their process could yield biodiesel at a selling price of $4.34 per gallon. They also conducted a sensitivity analysis to determine which parts of the process constitute the greatest costs, finding that the bottlenecks were algae cultivation and oil extraction operations. In particular, pond construction was the largest single cost and was also highly sensitive to location and climate. Clearly, there is an enormous amount of variation amongst these technoeconomic analyses, dependent upon the process design and other critical assumptions. While they are useful in estimating the economics of algal biofuel, they don't provide much insight into the sustainability of the processes.

Life cycle assessment is another important tool for analyzing biofuel production systems, providing key data such as energy return on investment (EROI) and greenhouse gas (GHG) emissions. Liu *et al* used this method in 2013 to evaluate the effectiveness of hydrothermal liquefaction as a process for biomass conversion (*166*). They modeled a full-scale commercial facility based on available pilot-scale data. They found that the modeled process yielded an EROI of about 2.5, compared to about 4.0 for petroleum. Although the EROI for their process is lower than petroleum, GHG emissions were reduced by approximately 60% compared to petroleum. The variety of cultivation, harvesting, and conversion technologies that are available for algal biofuel production complicates the assessment of both economic viability and environmental impact. One study attempted to tackle this complex landscape by comparing life cycle analysis results for multiple production pathways (*167*). The authors concluded that the cultivation and dewatering operations have a greater impact on GHG emissions than the conversion operations, highlighting the importance of selecting the appropriate technologies when constructing a biofuel production

pathway. Orfield *et al* produced the first life cycle analysis comparing photoautotrophic and heterotrophic algae cultivation methods (*168*). Although most analyses focus on photoautotrophic growth models (i.e. open ponds or photobioreactors), heterotrophic models (i.e. feeding sugars to algae) have also been proposed and tested. They found that the net energy ratio for heterotrophic growth is highly dependent on reactor performance and sugar source, but there is potential for heterotrophic growth to outperform photoautotrophic growth.

Mixotrophic growth is a model for algae production that combines heterotrophic and photoautotrophic growth models (i.e. providing organic and inorganic carbon sources). Kandimalla *et al* tested algae growth productivity under a mixotrophic model with flue gas and either glucose or sewage (*169*). They found that this strategy of growth was highly effective, with up to 85% removal of carbon dioxide from the flue gas. Additionally, they found that the algae could remove up to 75% of other nutrients and pollutants from the sewage, demonstrating the utility of algae in sewage treatment. Although they showed technical feasibility, they did not perform any economic or life cycle analyses for their process. Honda *et al* also demonstrated that algae can be successfully cultivated using treated sewage as a source of nutrients (*170*). They measured the rate of carbon dioxide absorption and tested different hydraulic and solids retention times, concluding that their methods enabled the highly efficient cultivation of microalgae, with 91% removal of nitrogen from the media. Alternatively, nutrients could be obtained from agricultural surpluses and waste sources, like pig and poultry manures, or byproducts of anaerobic digestion (*171*). However, major challenges face the utilization of such nutrient sources, including the large degree of variability across the different sources and the presence of pathogens or other contaminants. Nutrient recycling could help to mitigate input costs, as well as boost productivity in a closed photobioreactor system, as shown by Biller *et al* (*172*). Utilizing hydrothermal

liquefaction to convert algal biomass slurry into bio-crude, they recovered the aqueous fraction, determined its nutrient content, and added dilutions of this mixture back to the algal cultures, resulting in improved growth.

While many studies have focused on the production of algal biofuels, the concept of a biorefinery is built on the yield of an array of diverse products from algal biomass. In line with the biorefinery concept, Guarnieri and Pienkos reviewed the range of products that have been discovered in algae (*173*). They focused especially on the application of genomic, transcriptomic, proteomic, and metabolomic technologies for the systematic discovery of algal products. They argue that these technologies will enable new applications for algae biomass, which could add significant value and enable economical algae production processes. High-value products offer an important and complementary route to economic viability, as reviewed by Leu and Boussiba (*174*). Briefly, such high-value products include carotenoids, polyunsaturated fatty acids, and polysaccharides, for use as pigments and nutritional supplements. These biomaterials can be co-produced with biofuels from algae and are central to the concept of a biorefinery. In a demonstration of this concept, Dong *et al* developed a biorefinery process to co-produce sugars, lipids, and proteins from algal biomass (*175*). They utilized a dilute acid pretreatment to hydrolyze the biomass and release the various components, followed by fermentation, thermal treatment, and solvent extraction. Technoeconomic analysis of their process revealed that it reduced biofuel production costs by 9% compared to previous scenarios. Moreover, their process has potential to yield high-value co-products due to its nondestructive nature, but they did not take these into account in their analysis. The cumulative effect of all the research discussed above lends credence to the feasibility of commercial scale algae cultivation. However, it is also clear from this research

that more work is needed to bring down the production costs and find ways to obtain greater value from the biomass. In short, basic research on algae must be sustained.

1.3.2 Systems Biology and Metabolic Engineering

Systems biology, while recently emerging into the mainstream, has roots in the mid-20th century, as reviewed by Escosura *et al* (*176*). However, only recently have there been tools developed that are capable of testing hypotheses in systems biology. With the development of these tools, new ground has also been broken in the theoretical underpinnings of the field. This is enabling researchers to meaningfully probe the behavior of biological systems and obtain insights into functional mechanisms. A fundamental understanding of biological systems is required for successful metabolic engineering. Erb *et al* discuss the concept of synthetic metabolism, classifying engineering efforts into five levels (*177*). The most basic level consists of "copy, paste, and tune" within the constraints of naturally occurring pathways. The most advanced level consists of *de novo* enzyme and pathway design to construct artificial pathways with artificial enzymes. They discuss progress towards this vision and challenges that have been encountered so far along the way. Chubukov *et al* reviewed the challenges associated with applying the principles of synthetic and systems biology to commercial-scale production of chemicals via microbial processes (*178*). They cover a broad range of topics, including molecule selection, pathway design and construction, pathway optimization, toxic intermediates, host engineering, and scale up. They argue that if the challenges they discuss are successfully addressed, sustainable processes for bioproduction of commodity chemicals could achieve commercial viability. One major challenge is developing a "first principles" understanding of metabolic pathways based on their components. A powerful tool for achieving this goal is to utilize *in vitro* analyses. Lowry *et al* review eleven

examples of purely *in vitro* reconstitution of metabolic pathways (*179*). They trace the history of biochemistry through these examples, demonstrating the utility of *in vitro* analyses. In particular, they emphasize the determination of chemical mechanisms and the potential applications towards synthetic biology as well as traditional synthetic chemistry. Finally, they suggest that additional *in vitro* reconstitution studies of core metabolic pathways like nucleotide biosynthesis and the citric acid cycle could yield valuable insights into chemical mechanisms and kinetics. A challenge with this approach is the vast and overwhelming number of enzymes in existence. Despite this challenge, the proverbial ocean of enzymes also presents an opportunity. Guazzaroni *et al* review the concept of bioprospecting, wherein proteins from diverse microbes are screened for a desired catalytic activity (*180*). Briefly, they focus on a metagenomic approach to bioprospecting, which enables researchers to study microbes that cannot be cultivated in the laboratory. Considering that the vast majority of extant microbes cannot be cultivated in the laboratory, utilizing metagenomics broadens the horizon of possible proteins to screen. However, certain limitations, including poor expression of heterologous proteins and limited, non-optimal cloning vectors for library construction, currently present bottlenecks for enzyme discovery. They discuss how synthetic biology tools could help to alleviate these bottlenecks and improve the effectiveness of biocatalyst discovery with metagenomic screening.

Aside from microbes, humans have long depended on plants for nutrition, fiber, and medicine. New techniques in analytical biochemistry and metabolic engineering are transforming the ability to make use of plants, as reviewed by Wurtzel and Kutchan (*181*). They highlight how metabolic pathways that produce valuable compounds are being systematically discovered in plants and transferred to microbial systems. This requires a thorough understanding of the target pathways in plants. Derch *et al* reviewed advances in plant metabolic network analysis (*182*). They

discuss both experimental methods for obtaining data and theoretical models for data analysis and hypothesis testing. They emphasize the importance of isotopic labeling and pulse-chase experiments coupled with metabolic flux analysis, providing examples from the literature of the analytical power of this approach. Such experiments will continue to generate valuable new insights into plant physiology, enabling researchers to select and design favorable traits in plants. One trait that is a particularly important target is the ability to fix carbon dioxide. Erb and Zarzycki reviewed synthetic biology approaches to engineering improved photosynthetic fixation of carbon dioxide in plants and algae (*183*). They discuss four major strategies, focused on artificially optimizing RuBisCO, implementing carbon-concentrating mechanisms, engineering synthetic photorespiration bypasses, and designing synthetic carbon fixation pathways. Computational modeling of pathways plays an important role in the engineering process. Shi and Shwender reviewed the construction and utilization of genome-scale, constraint-based metabolic models for plants (*184*). They highlight experimental methods for testing the models, such as stable isotope labeling and tracer analysis to empirically determine metabolic flux. They emphasize that one of the major challenges facing plant metabolic models right now is the lack of experimentally validated enzyme functions and the reliance on predictions from databases. They suggest that the development of high-throughput methods for experimental determination of enzyme function will greatly improve model accuracy.

Numerous genome-scale metabolic models have been constructed for various organisms, from bacteria to eukaryotes. These models have proven immensely useful in elucidating physiological properties of the modeled organisms. Imam *et al* built a genome-scale metabolic model for the green microalga *C. reinhardtii* (*185*). The model consisted of 1,355 genes, 1,113 metabolites, and 2,394 metabolic reactions. They used the model to study the response to nitrogen

starvation, finding a concerted response to oxidative stress and priming for starch and lipid storage. Chapman *et al* utilized a previously published genome-scale metabolic model of *C. reinhardtii*, coupled with flux analyses, to determine the mechanism of photosynthetic repression during mixotrophic growth (*186*). In the presence of acetate, photosynthetic carbon assimilation is repressed, and cyclic electron flow mediates a bypass around PSI, mitigating electron flow from the oxygen-evolving complex. This enables the alga to increase the flux of carbon assimilation from acetate. Loira *et al* built a genome-scale metabolic model of the marine microalga *Nannochloropsis salina* (*187*). The model consisted of 934 genes, 1,985 metabolites, and 2,345 metabolic reactions, and made simple growth/no growth predictions on 32 different conditions with an accuracy of 90%. Furthermore, the model was able to predict growth rates with an average error of 15% for conditions with variable nitrogen sources and carbon dioxide levels. Levering *et al* built a genome-scale metabolic model for the diatom *Phaeodactylum tricornutum* (*188*). The model consisted of 1,027 genes, 2,172 metabolites, and 4,456 metabolic reactions, constrained by empirical observations of biomass composition (i.e. lipids, proteins, carbohydrates) obtained by Fourier transform infrared spectroscopy. The model enabled them to identify a previously unknown glutamine-ornithine shunt, which could play a role in transferring photosynthetic reducing equivalents to the mitochondria. Besides metabolic models, there are other types of genome-scale analyses that offer great scientific value. Wisecaver *et al* conducted a meta-analysis of global plant gene expression data, analyzing coexpression networks (*189*). They hypothesized that genes with a specialized metabolite pathway would form associations in the expression data, constituting coexpressed networks. They found that up to 52.6% of coexpressed gene modules contained two or more genes known to be involved in the biosynthesis of specialized metabolites.

They concluded that analyses of global gene expression networks provide a powerful tool for the discovery of biosynthetic pathways.

Systems and synthetic biology have substantial potential in terms of applications towards algae. Scaife and Smith reviewed recent progress in developing tools for algal synthetic biology (*190*). They suggest that combining advancements in transgene expression, genome editing, standardized genetic elements, and microfluidics, will improve not only the commercial potential of algae, but also the understanding of fundamental algal biology. Currently, one of the most important technologies for synthetic biology is the CRISPR-Cas9 genome editing system. In an early report on CRISPR/Cas9 manipulation of *C. reinhardtii*, Shin *et al* improved the mutagenic efficiency by direct delivery of Cas9 ribonucleoproteins (i.e. Cas9 coupled with gRNA) (*191*). Compared to vector-driven expression of Cas9 and gRNA, their approach reduced off-target effects and resulted in up to 100-fold greater mutagenic efficiency. Furthermore, they unexpectedly observed non-homologous end joining knock-in events, which provide novel opportunities for inserting DNA at a target locus. Preparing DNA fragments for genomic insertion is another area under active development. Recently, Shih *et al* reported a method for the assembly of large DNA fragments for engineering plant metabolism (*192*). They developed a novel "gene-stacking" method, taking advantage of yeast homologous recombination to assemble the fragments *in vivo*. The fragments can then be extracted from the yeast and utilized in *Agrobacterium*-mediated transformation of plants. This method is useful because it substantially reduces the amount of work required to assemble large DNA fragments, which are essential for synthetic biology. In addition to methods for DNA fragment assembly and genomic insertion, methods are needed for regulatory control of the constructs. Towards this goal, Liang *et al* developed a two-component system for the repression of transgene expression in plants (*193*). They made use of

the endoribonuclease Cys4 to target specific sequences in the 5'-UTR of the transgenic mRNA. This enabled them to repress transgene expression more than 400-fold and also to synchronize repression with certain molecular signals. They validated this system in both monocots and dicots, and demonstrated tissue-specific repression.

While plants and algae are important targets for synthetic biology, other organisms may provide greater utility. The yeast *Saccharomyces cerevisiae* is a good target for metabolic engineering because it is very well studied and highly genetically pliable. Tang *et al* reviewed efforts to engineer metabolism in this organism for increased production of fatty acids and fatty acid derivatives (*194*). They argue that although it has traditionally been used for alcohol production, rewiring its metabolism could make it highly valuable for lipid production. In this direction, Runguphan and Keasling engineered fatty acid biosynthesis in *S. cerevisiae* by overexpressing endogenous acetyl-CoA carboxylase and fatty acid synthase genes (*195*). This was achieved by replacing their native promoters with stronger constitutive promoters. The result was that the engineered strain accumulated lipid to over 17% of its dry cell weight, a 4-fold increase compared to the original strain. They also modified the final converting enzyme to yield free fatty acids and fatty alcohols, instead of triacylglycerols, which require less downstream processing. As with plants and algae, new tools are being developed to enable better genetic modification of *S. cerevisiae*. Apel *et al* developed a toolkit for engineering gene expression in *S. cerevisiae* using CRISPR-Cas9 (*196*). Their toolkit includes 23 Cas9-gRNA plasmids, 37 promoters with variable strength, and 10 tags to modulate protein localization, degradation, and solubility. To facilitate the use of their toolkit, they introduced a web-based application to assist researchers in the design of DNA fragments for genomic integration. Finally, they demonstrated the utility of their platform by optimizing the expression of taxadiene synthase, an important industrial enzyme, leading to a

25-fold improvement in taxadiene yield. Horwitz *et al* developed a method for multiplexed integration of genes via CRISPR-Cas9 genome editing (*197*). To test their method, they introduced into *S. cerevisiae* six DNA fragments, totaling 24 kbp and containing an 11-gene pathway for muconic acid biosynthesis. They achieved genomic integration with efficiency upwards of 64%, while also significantly reducing the time required in comparison to traditional methods. Although fatty acids are valuable, alkenes are even more desirable, as they are among the constituents of petroleum. Chen *et al* engineered alkene production in *S. cerevisiae* by incorporating fatty acyl decarboxylase enzymes and subsequent pathway optimization (*198*). Although they report their optimization efforts led to a 67.4-fold improvement in titer, they were only able to achieve a yield of 3.7 mg/L, which is not commercially relevant. Isoprenoids are another class of high-value compounds forming a target for industrial production with yeast. Meadows *et al* rewired the central carbon metabolism of *S. cerevisiae* to favor production of the building blocks for isoprenoids (*199*). By introducing four non-native enzymes, they obtained a strain that produced 25% more farnesene, while consuming 75% less oxygen during fermentation. Clearly, all of this work demonstrates that *S. cerevisiae* could serve as an excellent host for engineered pathways, perhaps consisting of enzymes extracted from plant genomes.

Although *S. cerevisiae* looks like a promising candidate, it does not naturally accumulate much lipid content, and significant optimization is required to redirect flux through this pathway. In contrast, *Yarrowia lipolytica* is an oleaginous yeast species with an inherent ability to accumulate large amounts of lipid, thus presenting an excellent model for engineering. Schwartz *et al* extended the CRISPR-Cas9 toolkit to *Y. lipolytica*, achieving remarkable transformation efficiency (*200*). They developed a synthetic promoter to express the CRISPR gRNA by combining native promoters for RNA polymerase III and tRNA. Using this promoter, along with

expression of a codon-optimized Cas9, and disruption of non-homologous end joining, they achieved 100% efficiency in markerless homologous recombination to integrate donor sequences into the genome at the target locus. In a demonstration of the genetic flexibility of this organism, Liu *et al* utilized an artificial evolutionary approach to engineer improved lipid production in *Y. lipolytica* (*201*). They developed a simple selection screen, coupled with mutagenesis, and conducted multiple rounds of selection. This led to a strain that had up to 87% lipid content and production titers of 39.1 g/L, an improvement of 55% over their previous strains. Whole-genome sequencing of the strain revealed a novel lipid production enhancer, a gene encoding a succinate semialdehyde dehydrogenase, pointing to a role for gamma-aminobutyric acid assimilation in lipogenesis. Heterologous expression of enzymes can also substantially affect lipid metabolism in *Y. lipolytica*. For example, Qiao *et al* used traditional molecular biology techniques to engineer lipid overproduction in the species (*202*). They identified a mammalian lipid regulatory associated with obese cellular phenotypes and expressed this gene in the yeast. They improved pathway flux by also overexpressing endogenous acetyl-CoA carboxylase and diacylglyceride acyltransferase genes. The resulting strain had a lipid production titer of approximately 55 g/L, the highest reported to date. Thus *Y. lipolytica* could also serve as a good host organism for heterologous and synthetic pathways prospected from other species.

In addition to the synthetic biology strategies discussed above, other studies have recently demonstrated significant advances in the ability to perform complex operations using genetic systems. Green *et al* developed an incredibly powerful "ribocomputing" device composed of RNA transcripts, which interact and enable computational logic operations (*203*). They assembled a ribocomputing circuit in a bacterium and demonstrated its ability to perform four-input AND, six-input OR, and complex 12-input combinatorial logic operations. Zalatan *et al* developed a novel

method for engineering complex transcriptional programs using a derivative of the CRISPR system (*204*). They fused regulatory protein binding motifs to the CRISPR guide RNA (so-called scaffold RNAs), enabling specific, targeted regulatory functions for a given locus. Furthermore, they were able to engineer multiplexed scaffold RNAs to target several loci simultaneously, each with different regulatory actions. In a demonstration of their method, they engineered a highly branched metabolic pathway in yeast, allowing them to selectively program pathway flux. Bradley *et al* reviewed the challenges associated with adapting natural genetic components to build synthetic digital-like circuits for biocomputation (*205*). They discussed recent advances in the development of standardized part families, which form the basis of genetic logic circuits. Finally, they suggest that improvements to these parts are necessary to enable signal fidelity in deeply layered circuits. These technologies will continue to yield advancements in the field of synthetic biology, allowing researchers to design, build, and test more complex biological systems. As the methods become standardized and the costs become lower, sustainable biotechnology will begin to have a larger impact and reduce the environmental impact of human civilization.

2. THE GENOME OF *BOTRYOCOCCUS BRAUNII*

The following sections describe the process of assembling the *B. braunii* genome, which was carried out over the course of several years. It begins by reviewing DNA sequencing technologies and computational algorithms for genome assembly. After that, experimental data are presented on the performance of various tools for assembling the *B. braunii* genome sequencing data. Finally, the methods used to build the current assembly are described in detail, as well as the methods that were utilized to annotate the assembly with various genetic features.

2.1 Introduction

Genome assembly is a constantly and rapidly evolving field of science. The following section attempts to provide a comprehensive overview of the genome assembly process and discusses some of the most recent advances in the field. This is essential context for understanding all of the work on assembling the *B. braunii* genome, which is described in the subsequent sections.

2.1.1 DNA Sequencing Technologies

Tools for genome sequencing have evolved enormously since they were first invented in the 1970s (*206*). Since the turn of the 21st century especially, the evolution of DNA sequencing has rapidly intensified (*207*). Briefly, three major sequencing platforms emerged in the early 2000s: 454 Life Sciences pyrosequencing, Illumina sequencing-by-synthesis, and Pacific Biosciences (PacBio) single-molecule real time (SMRT) sequencing. While 454 Life Sciences was initially successful, it was quickly overtaken by Illumina, which continues to dominate the market. PacBio was a smaller player until about 2012, at which point it began to grow steadily. Although Illumina is still the market leader, PacBio technologies have made substantial improvements to

genome sequencing and its market share continues to grow. The pace of development gives no indications of slowing down and further advancements in the field are just around the corner (*208*). With every new sequencing technology that is invented, new challenges arise in processing and analyzing the data (*209*). Moreover, no sequencing platform is perfect, and there is evidence of platform-specific and species-specific biases that arise in genome data (*210*). In particular, one study analyzed the effect of systematic errors and random errors on the determination of genomic variants (*211*). The authors argued that Illumina suffers from systematic errors, whereas PacBio has random errors, providing a major benefit because random errors can be corrected by obtaining a sufficiently large sample of sequence data (i.e. high coverage, or deep sequencing). The challenges of assembly will be further discussed in the following sections.

2.1.2 De Novo Genome Assembly Tools

The utility of DNA sequencing data is the ability to reassemble it into sequences approximating the true genomic sequence. Assembly is a complex task and is performed by computer programs that are designed to handle certain, specific data. Broadly, there are two major classes of genome assembly algorithms: overlap-layout-consensus (OLC) and de Bruijn graph (DBG) approaches (*212*). Virtually all of the early assembly programs were OLC-based, but among them ARACHNE, released in 2002, stands out as the first program designed specifically to handle "paired end" data (*213*). The DBG-based approach appeared in 2001, with the release of EULER by Pevzner *et al* (*214*). The first wave of assemblers for next-generation sequencing data was thoroughly reviewed in 2010 by Miller *et al* (*215*), including SSAKE, VCAKE, Newbler, Celera, ARACHNE, CAP, EULER, Velvet, ABySS, AllPaths, SOAPdenovo, and others. While they summarized the functions of these different programs, they made no assessment or

comparison of their quality. However, it is important to determine which programs are superior in performance and why, so as to enable the design of better programs in the future. In 2011, results from a competition called Assemblathon 1 were released, demonstrating the effectiveness of different assemblers, handled by volunteer teams, all of them assembling a standardized set of data (*216*). Unfortunately, there was not a conclusive result in terms of which was the best performing assembler. They found that there was an enormous amount of variability in the assemblies that were submitted to the competition. Furthermore, the Assemblathon made clear that comparing genome assemblies is quantitatively and qualitatively difficult, at least in part due to a lack of statistical measures to describe completeness, quality, and correctness. That same year, Narzisi and Mishra (*217*) attempted to introduce a more comprehensive metric for assessing genome quality called the Feature-Response Curve. Another approach to assessing the completeness and quality of a genome assembly is to look for the presence of highly conserved genes. Simao *et al* (*218*) developed a program in 2015 called BUSCO (Basic Universal Single Copy Orthologs) that scans genomes, transcriptomes, or proteomes, to determine the fraction of BUSCOs recovered. Despite the challenges with quality assessment, the development of genome assembly software has continued at a blistering pace over the last five years. Developers have largely focused on algorithms for Illumina and PacBio sequencing data, currently the two main technologies used for sequencing genomes.

Following is a selection and summary of significant advancements since 2012 in genome assemblers designed for Illumina data. Simpson and Durbin (*219*) introduced SGA (String Graph Assembler), utilizing for the first time a FM-index derived from a Burrows-Wheeler transform and implementing a string graph in the assembly process. This enabled a highly compressed representation of the reads, which is important because the large size of such datasets is a major

challenge what with the limited availability of computational memory. Their assembler was oriented towards large eukaryotic genomes. Bankevich *et al* (*220*) introduced SPAdes, an assembler oriented towards bacterial genome assembly, with a special focus on assembling single-cell sequencing datasets. They developed new DBG-based data structures and graph theoretical approaches for information processing and genome reconstruction. The challenge of single-cell sequencing in particular is a large amount of coverage bias in the data. Zimin *et al* (*221*) introduced MaSuRCA, an assembler that combines DBG and OLC approaches. It uses a DBG to construct "super reads" from the original reads and then uses an OLC algorithm to assemble the super reads. This enables the assembler to combine sequencing data with variable read lengths (i.e. Illumina with 454 or Sanger). They found that MaSuRCA was flexible enough to assemble a diverse range of genomes, from bacteria to large and complex eukaryotes, with quality on par with or better than existing assemblers such as ALLPATHS-LG and SOAPdenovo2. Weisenfeld *et al* (*222*) introduced DISCOVAR, which started out as a variant calling algorithm but evolved to include *de novo* assembly capabilities. This piece of software is unique in that it is specifically designed to accept an Illumina 2x250 bp paired end library with a fragment size of 400-800 bp, prepared with a PCR-free protocol. Although it is inflexible in terms of data input, DISCOVAR does an incredible job of assembling the data it was designed for, generally giving very good results. Chikhi *et al* (*223*) introduced BCALM2, a novel algorithm for DBG construction and compaction. They developed a minimizer hashing technique that reduces computational memory consumption and run time by approximately an order of magnitude. This is very important because Illumina sequencing datasets for eukaryotic genomes are typically quite large and computational memory requirements to handle this data have become a limiting factor. Another approach to reducing the memory required for data handling is to use a probabilistic data structure called a Bloom filter.

Jackman *et al* (*224*) introduced ABySS 2.0, implementing a Bloom filter and building on the previous versions of the well-established assembler. Their new implementation could assemble a human genome dataset with less than 35 GB of memory, while maintaining assembly quality compared to previous benchmarks.

Following is a selection and summary of significant advancements since 2012 in genome assemblers designed for PacBio data. One of the earliest assembly programs for pure PacBio data was HGAP (hierarchical genome-assembly process), introduced by Chin *et al* (*225*). The algorithm utilized a directed acyclic graph to represent read overlaps and find contiguous paths through the graph, resulting in raw, low-accuracy genomic sequences. The contiguous sequences (contigs) were then corrected with a consensus algorithm to reach over 99.999% accuracy. However, this work was conducted using the bacteria *E. coli*, which does not have a very high degree of genomic complexity. By contrast, eukaryotic genomes typically have a substantial amount of repetitive DNA elements, severely impeding resolution of the sequences. To deal with highly complex repeats, Kamath *et al* (*226*) introduced HINGE, an algorithm to achieve optimal repeat resolution when assembling long reads (i.e. PacBio). The program was so named because of how it added "hinges" to the reads while constructing an overlap graph, enabling the differentiation of resolvable and unresolvable repeats. Polyploidy is another challenge that is faced by genome assembly algorithms. This issue was specifically addressed in the FALCON program, introduced by Chin *et al* (*227*) as a successor to HGAP. FALCON is an OLC-based assembler that maintains awareness of diploid haplotypes during the assembly process. Then a companion program called FALCON-Unzip resolves the haplotypes. They demonstrated this approach by assembling several eukaryotic genomes, finding accurate phasing of the haplotypes. Koren *et al* (*228*) introduced Canu, a successor of the Celera Assembler, to address both repeats and ploidy. The OLC-based

algorithm makes use of adaptive k-mer weighting and repeat separation strategies to yield high-quality assemblies. Furthermore, the program preserves the graph structure in the output, enabling further analyses. While all of the aforementioned assemblers for PacBio were based on an OLC approach, Lin *et al* (*229*) introduced ABruijn, the first long-read assembler to incorporate a DBG-based approach. However, the ABruijn algorithm still makes partial use of an OLC-based approach. Nonetheless, the program was able to generate high-quality assemblies and represented an important advance in the field. Most recently, Kolmogorov *et al* (*230*) introduced Flye, a successor to ABruijn, which takes a novel approach to the assembly graph construction. The algorithm first generates inaccurate overlapping contigs and then recombines these contigs into an accurate assembly graph that represents the repeats in a manner consistent with the reads. This graph is then used to find paths, which are output as the final accurate contigs.

While many scientists have focused on developing algorithms for either Illumina or PacBio data, some have developed hybrid approaches that combine both data types. Deshpande *et al* (*231*) introduced Cerulean, a program to resolve repeats by mapping long reads to an assembly graph generated with short reads. However, this program was poorly tested, incomplete, and essentially useless from a practical standpoint. Ye *et al* (*232*) introduced DBG2OLC, which utilizes contigs from a DBG-based short read assembly to compute anchor points in long reads and produce an OLC-type assembly from the long reads. Their program is well tested and fully functional, but does not offer substantial improvements over pure-PacBio assemblers when there is sufficient sequence coverage. Zimin *et al* (*233*) introduced a hybrid version of MaSuRCA, capable of utilizing both Illumina and PacBio reads in the assembly process. Another approach to obtaining hybrid genome assemblies is to consolidate separately assembled sequencing datasets. Scholz *et al* (*234*) introduced MeGAMerge, a pipeline for metagenome assembly by combining contigs from

assemblies of short and long reads. Along similar lines, Wences and Schatz (*235*) introduced Metassembler, a pipeline to merge multiple assemblies of a single genome into one super-assembly, with the most accurate sequences from each of the sub-assemblies. The idea of assembly consolidation is compelling, but faces many challenges (*236*). These tools are highly dependent on the quality of the input assemblies, and do not perform consistently on standardized datasets. Thus much work remains to be done in developing not only better *de novo* assembly tools, but also downstream tools for assembly processing and consolidation to generate high-quality genome sequences.

2.1.3 Mapping DNA Reads to the Genome

Almost all of the steps downstream of *de novo* assembly are built on the requirement of aligning sequence reads back to the genome assembly. Some of the earliest algorithms for the alignment of short reads were reviewed in 2010 by Li and Homer (*237*). In the two years leading up to their review, over 20 short read aligners were published. They discuss a number of these aligners in great detail, giving consideration to both the alignment theories and computational performance. Significantly, they conclude that although short reads dominated at the time, and the community of researchers developed an extensive set of aligners, in a few years long reads would dominate again and new aligners will be needed. More recently, in 2015, Reinert *et al* (*238*) reviewed the alignment of next generation sequencing reads. They begin by providing an overview of the available sequencing technologies and their associated errors. Instead of a survey of published aligners, they deeply discuss the underpinning algorithmic approaches used in designing alignment programs. They suggest that future advancements may take advantage of graphics processing units (GPUs) or coprocessors like the Intel Xeon Phi. Despite the advances in

algorithms for approximate pattern matching in strings, developing better alignment programs will require a better fundamental understanding of genome sequences. For example, Li *et al* (*239*) examined the k-mer frequency distributions for k ranging from 20 bp to 1,000 bp in the human genome. They found that the ability to uniquely map reads to the genome gives diminishing returns when the read length exceeds 200 bp and even a read length of 1,000 bp was insufficient to uniquely map all reads. Moreover, different species have different genomic properties and will thus have different effects on the mapping process. The difficulty originating from the incredible amount of sequence diversity is compounded by the lack of well-defined benchmarks and standards for mapping algorithms. Smolka *et al* (*240*) attempted to address this problem by introducing Teaser, a program to automatically benchmark different aligners and parameter settings for a given dataset. The goal of this program is to enable researchers to quickly assess which aligner and what parameters are optimal for their conditions.

Although a myriad of read mapping programs are already available, the pace of new developments has not slowed. In 2012, Chaisson and Tesler (*241*) introduced BLASR (basic local alignment with successive refinement), a new paradigm for aligning PacBio reads to a genome. This program is the official aligner of PacBio and is specially built to handle the raw PacBio data. In 2015, Kim *et al* (*242*) introduced HISAT, which combined the Burrows-Wheeler transform with the Ferragina-Manzini index to yield a short read aligner with unprecedented speed, while also maintaining low memory consumption and very good accuracy. In 2016, Liu *et al* (*243*) introduced deBGA, a short read aligner that utilizes a DBG-based approach to mapping the reads. The authors claim it is faster, more sensitive, and more accurate than other state-of-the-art short read aligners. That same year, Sovic *et al* (*244*) introduced GraphMap, an algorithm designed for long reads with potentially high error rates (e.g. PacBio). They claim that compared to other aligners, GraphMap

gives a 10-80% increase in mapping sensitivity. The following year, Deorowicz *et al* (*245*) introduced Whisper, an aligner that first sorts reads and then maps them against suffix arrays of the genome. Also in 2017, Li (*246*) introduced Minimap2, an aligner which makes use of a minimizer hashing technique to dramatically increase alignment speed and is also capable of handling long, noisy reads, as well as short reads, or assembled contigs. It is likely that new alignment paradigms will continue to emerge in the coming years, posing new challenges to those who seek to align sequencing reads to genomes. The primary challenges being to systematically assess aligner performance and minimize false alignments.

2.1.4 Scaffolding, Gap Filling, and Polishing

Once the sequencing reads have been aligned back to the genome assembly, a number of important operations can be executed. When it comes to further improving the quality of the genome assembly, the primary operations of interest are scaffolding, gap filling, and polishing (*247*). Scaffolding consists of orienting and ordering the contigs in an assembly into linear groups of contigs, separated by gaps of unknown sequence (*248*). This process is achieved by aligning paired end reads (i.e. mate pairs), obtained by sequencing both ends of a large DNA fragment. When the mates align to different contigs, it establishes a link between those two contigs, allowing them to be placed into a scaffold. The resulting scaffolds contain numerous gaps of unknown sequence, represented by the character "N" for an ambiguous nucleotide. These gaps could contain important sequences and so it is important to try and determine what those sequences are by filling the gaps (*249*). Finally, there might be errors in the assembly, which need to be corrected by polishing the sequences (*250*). Following are a selection and discussion of tools for achieving each one of these critical processes.

In 2014, Hunt *et al* (*248*) thoroughly reviewed genome scaffolders, including the stand-alone tools Bambus2, GRASS, MIP, Opera, SCARPA, SOPRA, SSPACE, and the scaffolding modules from ABySS, SGA, and SOAPdenovo2. Their study was complicated by the multitude of aligners available for mapping the reads to the genome, and they found that the selection of aligner had a significant impact on the scaffolding results. Nonetheless, they conclude that SGA, SOPRA, and SSPACE generally yield the best performance, given their datasets. That same year, two additional scaffolding tools were published. Sahlin *et al* (*251*) introduced BESST, designed to scaffold genomes of all sizes and complexities. They found that their algorithm performed well compared to some of the other available scaffolders, especially when the fragment size distribution of the library has a large standard deviation. This feature is particularly important, because for libraries with large fragments, it is very difficult to obtain a narrow size range of fragments, though new tools such as BluePippin are yielding better results (*252*). In contrast to the paradigm of scaffolding with mate pair libraries, Boetzer and Pirovano (*253*) introduced SSPACE-LongRead, designed to scaffold assemblies using PacBio long reads. They tested their program on six bacterial genome assemblies generated from short read data. They concluded that their algorithm could reliably and quickly scaffold bacterial genomes, but it is difficult to say how well this conclusion extends to more complex eukaryotic genomes. In 2015, Warren *et al* (*254*) introduced LINKS, another scaffolder designed to utilize long reads. Their approach was unique in that it is an alignment-free method, utilizing instead k-mer information content to determine scaffolds. This frees it from the issues associated with selecting an alignment algorithm for the reads. Moreover, they applied LINKS to the *S. ceveresiae* genome, demonstrating its utility for eukaryotes.

Gap-filling algorithms have also evolved to make use of either PacBio or Illumina data. In 2012, Boetzer and Pirovano (*255*) introduced GapFiller, designed to close gaps using paired reads

and an OLC-based assembly process. They tested the algorithm on both bacterial and eukaryotic datasets, obtaining good results with few errors. At the same time, English *et al* (*249*) introduced PBJelly, the first gap filler designed to make use of PacBio long reads. They tested the algorithm on a wide range of eukaryotic genomes, finding that it could close upwards of 90% of gaps in some assemblies. However, they also found it has a propensity to overfill or misassemble some gap sequences, which could reduce the quality of the assembly. In 2014, Piro *et al* (*256*) introduced FGAP, an algorithm capable of handling several different data types, including short reads, long reads, and preassembled contigs. They found that it could reduce the number of gaps by 78% in an *E. coli* assembly and by 35% in a human chromosome assembly. However, the filled gaps were not extensively evaluated for misassemblies and thus it is unclear exactly how reliable this tool is for accurately filling gaps. In 2015, Paulino *et al* (*257*) introduced Sealer, with a DBG-based gap assembler that utilizes a Bloom filter data structure, as part of the ABySS toolkit. This gap filler was specifically designed to be scalable to very large genomes and sequencing datasets. They tested it on the human and white spruce draft genome assemblies, finding that it could close 50.8% and 13.8% of gaps, respectively.

While gap fillers can create assembly errors, the *de novo* assemblers themselves can also create errors. In order to obtain a high-quality draft genome sequence, it is important to correct these errors, which can be done with either Illumina or PacBio data, with several tools available to do so. In 2012, Ronen *et al* (*258*) introduced SEQuel, designed to correct insertion, deletion and substitution errors in assembled contigs. This algorithm models the correct sequences with a DBG-based approach. When applied to an *E. coli* draft assembly, it reduced by nearly half the number of small insertions and deletions, and corrected 30-94% of substitution errors. In 2013, when Chin *et al* (*225*) published the HGAP assembler, they also introduced Quiver, a polishing algorithm for

PacBio data. With sufficient coverage, this algorithm is able to generate very high-quality sequences, exceeding 99.999% accuracy. In 2014, Walker *et al* (*259*) introduced Pilon, designed to accept paired end reads from small and large fragments. This algorithm is capable of correcting errors as well as filling small gaps. In 2017, Vaser *et al* (*260*) introduced Racon, designed to generate consensus sequences from long reads to obtain high-quality assemblies. They found that their algorithm determined a consensus more than an order of magnitude faster than other available methods. However, all of these methods for error correction are dependent upon first aligning the reads to the genome, which impacts the results of error correction. Thus further work is still needed to determine best practices for error correction.

2.1.5 Summary of the B. braunii Version 1.0 Genome

Almost three years to the day after HudsonAlpha delivered the first draft assembly (Version 0.5, built with ARACHNE) of the *B. braunii* genome, the "Version 1.0" draft assembly was completed and prepared for release (Table 1). The "Version 0.5" draft assembly (December 2013) consisted of 1,644 sequences with a N50 value of 287,289 bp, for a total of 147.0 Mbp at 50.33% GC content and 15.6% gap content. The "Version 1.0" draft assembly (December 2016) consisted of 2,752 sequences with a N50 value of 372,998 bp, for a total of 184.4 Mbp at 50.83% GC content and 2.5% gap content. This assembly was mainly based on FALCON contigs assembled by the team members at HudsonAlpha, but did incorporate some of the DISCOVAR contigs presented below. Following is a description of the methods used to generate the "Version 1.0" draft assembly.

The PacBio data was assembled with FALCON-UNZIP and resulting sequences were polished using QUIVER. To detect misassemblies, the library LCHA was aligned to the sequences and fragment coverage over each base was computed. A drop in fragment coverage below 10X

indicated an assembly error. By this measure, 19 assembly errors were detected. These errors were removed by breaking the assembly at the indicated regions. Separately, the Illumina library SXPX was assembled with DISCOVAR *de novo*. Sequences were identified in the DISCOVAR assembly that were not present in the FALCON assembly. This was achieved by masking the DISCOVAR assembly using 24-mers from the FALCON assembly. Regions greater than 2 kb that were not masked by this process were then extracted from the DISCOVAR assembly. A total of 487 unmasked regions containing 1.396 Mbp of sequence were extracted. These sequences were combined with the broken FALCON assembly and scaffolded using SSPACE with the library LCHA. Finally, the assembly was error-corrected using a sample of reads from the Illumina library SXPX, giving about 42X coverage over the genome. Analysis revealed 523 scaffolds (19.8 Mbp) that did not share a significant number of 24-mers with the rest of the assembly. These sequences were queried against the NCBI non-redundant database, identified as prokaryotic contamination, and removed from the assembly. Mitochondrial and chloroplast sequences were removed prior to assembly based on the published sequences for these organelle genomes (*136*).

Table 1. Statistics of *B. braunii* genome v0.5 and v1.0 assemblies. The v0.5 assembly had much fewer bases than the expected genome size and only three sequences larger than a megabase. The v1.0 assembly had more bases than the expected genome size, and much fewer gaps than v0.5. However, there were still a low number of large fragments recovered in the v1.0 assembly.

	Version 0.5 Assembly	Version 1.0 Assembly
# contigs (>= 0 bp)	1,644	2,752
# contigs (>= 10 kbp)	894	998
# contigs (>= 100 kbp)	458	477
# contigs (>= 1 Mbp)	3	4
Total length (>= 0 bp)	147,010,953	184,385,342
Total length (>= 10 kbp)	145,339,737	178,062,322
Total length (>= 100 kbp)	127,003,321	159,204,275
Total length (>= 1 Mbp)	3,386,438	5,632,961
Largest contig	1,245,454	1,870,169
GC (%)	50.33	50.83%
N50	287,289	372,998
L50	157	156
# N's per 100 kbp	15,645	2,501

2.2 Materials and Methods

This section describes the experimental procedures implemented in the course of data generation, processing, and analysis.

2.2.1 Biological Materials and Methods

The *B. braunii* genome project began at about the same time as the first next-generation sequencers were becoming commercially available. In 2010, the Joint Genome Institute (JGI), part of the US Department of Energy (DOE), approved a project proposal to sequence and assemble the *B. braunii* race B (Showa strain) genome. This was a follow-up to a project that JGI had approved a year prior, to sequence and assemble expressed sequence tags (ESTs) for the species. While the *B. braunii* EST project is not the focus of this work, it is worth mentioning that it resulted in 495,985 pyrosequencing reads containing 71.6 Mbp of sequence data, produced with a 454 GS FLX Titanium sequencer. Using this same technology, the first genome sequence data was generated using genomic DNA (gDNA) isolated from *B. braunii* race B (Showa), for a total of 16,542,544 pyrosequencing reads with 4.8 Gbp of sequence data, yielding approximately 29X coverage on the genome. The mean read length was 293 bp, with minimum 40 bp and maximum 1,196 bp, although the vast majority of reads were below 600 bp in length (Figure 1). While this is a substantial amount of data, it pales in comparison to what would be generated with Illumina and PacBio technologies. Furthermore, the software available to assemble 454 data never really matured, making it difficult to work with this data.

By 2012, Illumina had overtaken 454 as the leader in sequencing technology, and just one year later, 454 would go out of business. Around this time, the JGI constructed and sequenced several libraries of *B. braunii* gDNA using the Illumina platform (Table 2). The libraries NGNB

and HOOW were finished in 2012-2013, library SXPX was finished in 2014, and library LCHA was finished in 2016-2017. Library SXPX is special because it was constructed with a PCR-free protocol and sequenced with the 2x250 bp chemistry, enabling it to be assembled with a unique algorithm, which will be discussed in the following sections. Libraries NGNB and HOOW were mate pair libraries made with the transposon method, constructed and sequenced by the JGI. Library LCHA was made by Lucigen Corp., a private biotechnology company, constructed with the NxSeq method, and then sequenced partially by Lucigen and partially by HudsonAlpha, a private institute and collaborator with the JGI. The mate pair libraries, in particular LCHA, were crucial for the assembly effort, as they are necessary for effective scaffolding, which will be discussed later. The end result, however, is the ability to assemble larger pieces of DNA, as well as assess errors and structural variants.

As the last rounds of Illumina sequencing were being completed, the JGI initiated PacBio sequencing efforts for the *B. braunii* genome. In 2015-2016, two libraries were constructed and sequenced across 54 separate SMRTcells on the PacBio RS II platform with the P6-C4 chemistry. The result was 7,425,977 reads containing 36.3 Gbp of sequence data, yielding approximately 219X coverage on the genome (Figure 2). For the total read set, the mean read length is 4,892 bp, with a standard deviation of 3,419 bp. With PacBio data, it is highly desirable to have very long reads, and excluding reads shorter than 4 kbp (peak bin in read length histogram) reduces the number of reads to 4,334,642 but increases the mean read length to 6,883 bp, while the coverage remains quite high (~170X). Although these reads are very long compared to Illumina reads, there is rather low coverage with PacBio reads that are greater than 10 kbp in length, making it hard to compete with the high coverage of library LCHA (fragment size 15 kbp).

The availability of so much different sequencing data for the *B. braunii* genome not only enables a high-quality genome assembly, but also makes it a very interesting case study for comparing the efficacy and quality of different sequencing platforms. The challenge in this scenario is how to make the best use of these data. The answer is highly dependent upon the availability of software to support assembly and subsequent analyses.

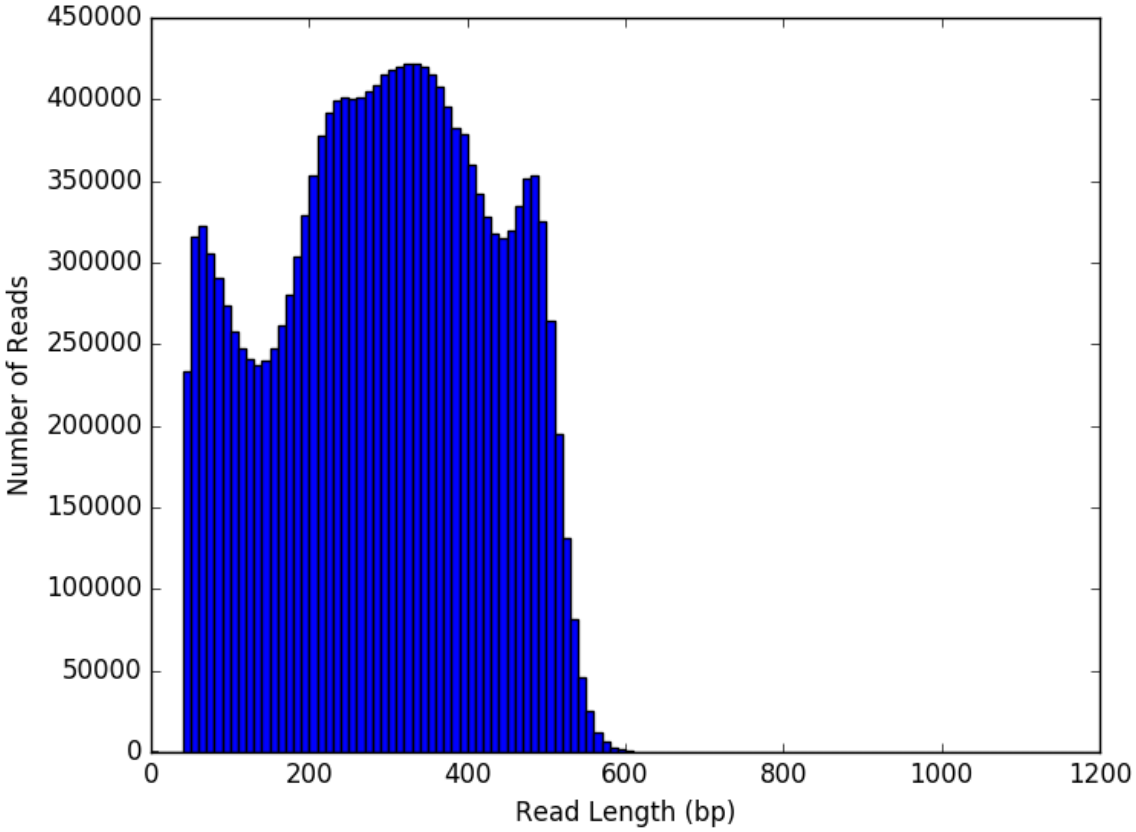


Figure 18. Summary of 454 Life Sciences pyrosequencing data obtained for *B. braunii.* The majority of reads were of length between 200 and 500 bp. There were 16,542,544 total reads containing 4.6 Gbp of sequence data, giving approximately 29X sequence coverage of the *B. braunii* genome.

Table 2. Summary of Illumina paired-end sequencing data obtained for *B. braunii.* The four libraries used on this work were constructed over a period of several years, from different samples of *B. braunii* gDNA. The inconsistency in samples used throughout library preparation adds to the challenges of assembly and analysis.

Library Name	Mean Fragment Size (bp)	Library Size (Total Pairs)	Paired Read Length (bp)	Sequence Coverage	Fragment Coverage
SXPX	762	249,536,701	2x250	752X	1,202X
NGNB	1,855	144,166,620	2x150	261X	1,302X
HOOW	4,649	100,488,168	2x150	182X	2,421X
LCHA	15,671	17,138,871	2x300	24X	1,549X

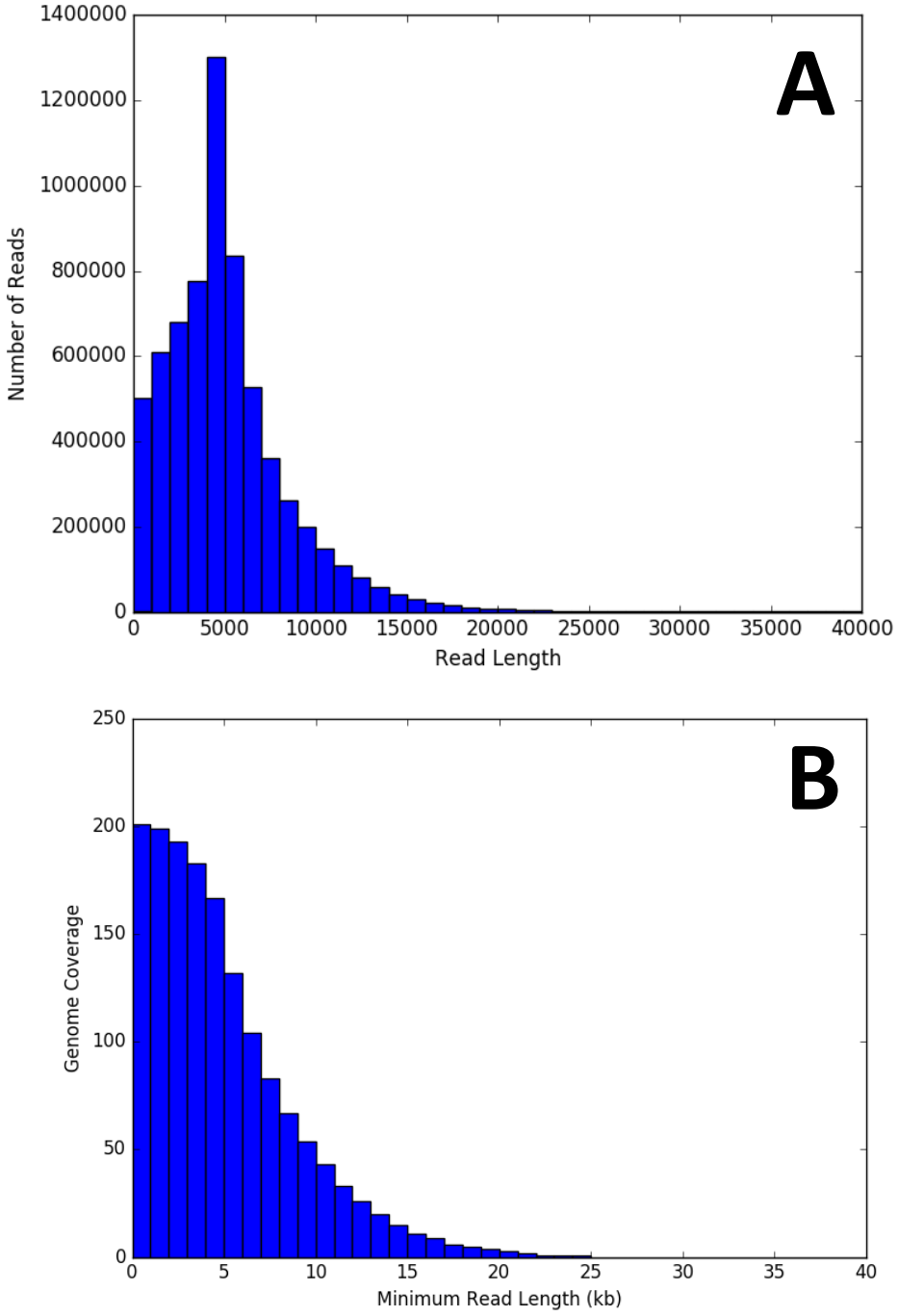


Figure 19. Summary of Pacific Biosciences sequencing data obtained for *B. braunii.* There were 7,425,977 reads and the majority of them were below 10 kb in length. This is undesirable, as longer reads help resolve complex genomic repeats and result in better assemblies. However, the coverage of the data is quite substantial, which is very important for consensus base calling in the assembled sequences.

2.2.2 Computational Materials and Methods

All of the computational materials and methods utilized in the course of this work are described in Appendix A.

2.3 Results and Discussion

This section presents the resulting data and provides discussion of the quality, impact, and interpretation of the information obtained.

2.3.1 Testing Assembly of the B. braunii Genome

The following sections describe the earliest efforts to assemble the *B. braunii* genome, which resulted in a valuable collection of lessons that were applied in subsequent efforts to assemble genome. The data presented below are generally valuable to anyone who is working to assemble a complex eukaryotic genome and offer insights into the fundamentals of the assembly process, prompting many questions about how to best assemble a genome.

2.3.1.1 Combining Multiple de Bruijn Graph Assemblies

DBG-based assemblies are based on the extraction from sequencing reads of DNA strings of length k, called k-mers (*261*). A graph is constructed from the k-mers based on overlaps of k–1 bases (*261*). In the ABySS assembly process, unitigs are the unbranching paths in the graph (*262*). There are certain features of the graph called tips and bubbles, which result from allelic differences, single nucleotide polymorphisms, sequencing errors, etc (*262*). The most important choice for a DBG-based assembly is the selection of k-mer size, and much debate has focused on what is the optimal value of k for an assembly (*263*). While there is no clear answer to this question, it is clear that the upper limit of k is the length of the reads. In order to assess the impact of k-mer size on

assembly contiguity, the *B. braunii* library SXPX was assembled with ABYSS across a range of k-mer values, from 50 to 250 (Figure 3). This experiment revealed some very interesting patterns, with an apparent optimum for the assembly N50 value at a k-mer size of approximately 125, or half of the read length (250 bp). The total assembly size and the maximum unitig size also show apparent maximums at this k-mer value.

Considering that an individual assembly is one singular model of the genome, it is possible that no single assembly will optimally capture all of the genomic sequence, especially in a DBG-based approach to assembly. It was hypothesized that different values of k-mer may optimally assembly different regions of the genome, and that in order to obtain the whole genome sequence, multiple assemblies with different k-mers must be consolidated (Figure 4). To test this concept, CD-HIT (*264*) was employed to de-replicate the unitigs from ABYSS (Figure 5). Across all of the assemblies, 96.9 million unitigs were assembled. Of these, 23.2 million (or 24%) were equal to or greater than 300 bp in length, while 8.4 million (or 9%) were equal to or greater than 1,000 bp in length. Both of these length thresholds were used to create selections of the total unitigs for de-replication. CD-HIT was used to de-replicate the sequences with identity thresholds of 100%, 99%, and 95% (Table 3). Further consolidation of the assembled sequences was achieved by employing an OLC approach using programs from the ABySS toolkit. The set of sequences with a minimum length of 1,000 bp, de-replicated by CD-HIT with a 100% identity threshold (called SEQ1K 100) had a total of 390.8 Mbp of sequence, and after OLC re-assembly was further consolidated to 215.0 Mbp of sequence (Table 4). Despite the successful consolidation, the contiguity of these sequences remains quite low, with a total of 77,713 sequences and a N50 value of 3,342 bp. These experiments clearly demonstrate that a "k-mer scanning and consolidation" approach to assembling a genome can provide more value than a single k-mer assembly.

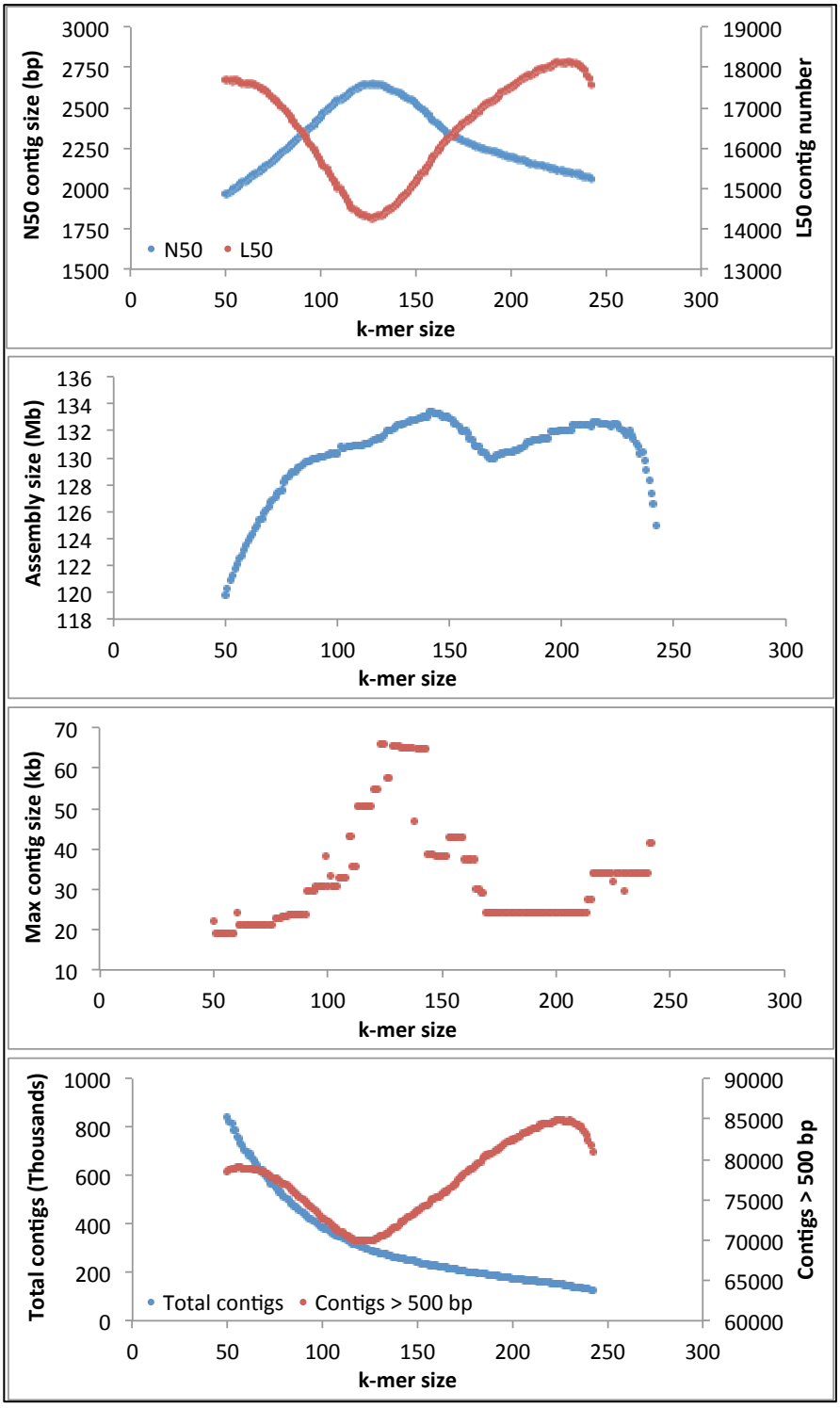


Figure 20. Assembly statistics of ABYSS at different k-mer values. Using the Illumina library SXPX, 192 assemblies were generated with ABYSS, each with a different k-mer setting. The range of values for k was 50 to 242. The above visualizations of the assembly statistics show interesting patterns, with an apparent optimum when k equals approximately half the read length.

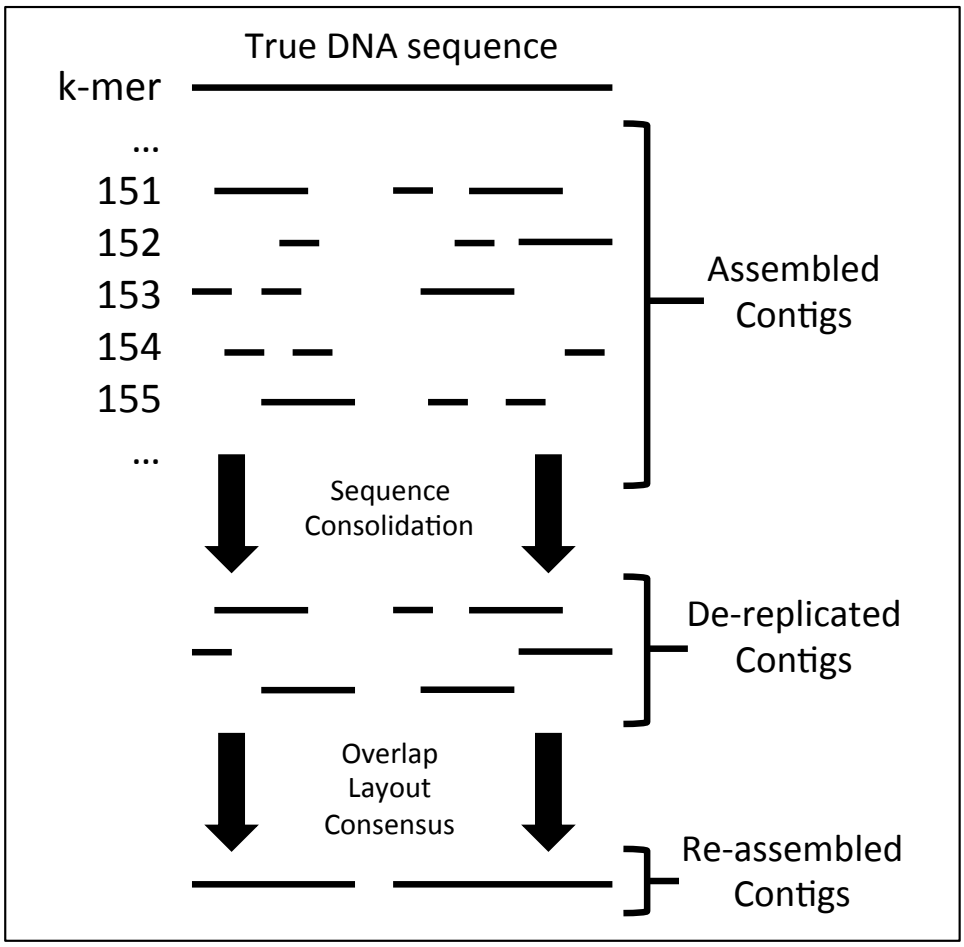


Figure 21. Overview of concept to combine multiple sub-assemblies. The idea behind combining multiple different sub-assemblies is that each sub-assembly captures unique and non-unique elements of the genome. A single sub-assembly is incomplete in its information content, but across all of the sub-assemblies, a more complete model of the genome emerges.

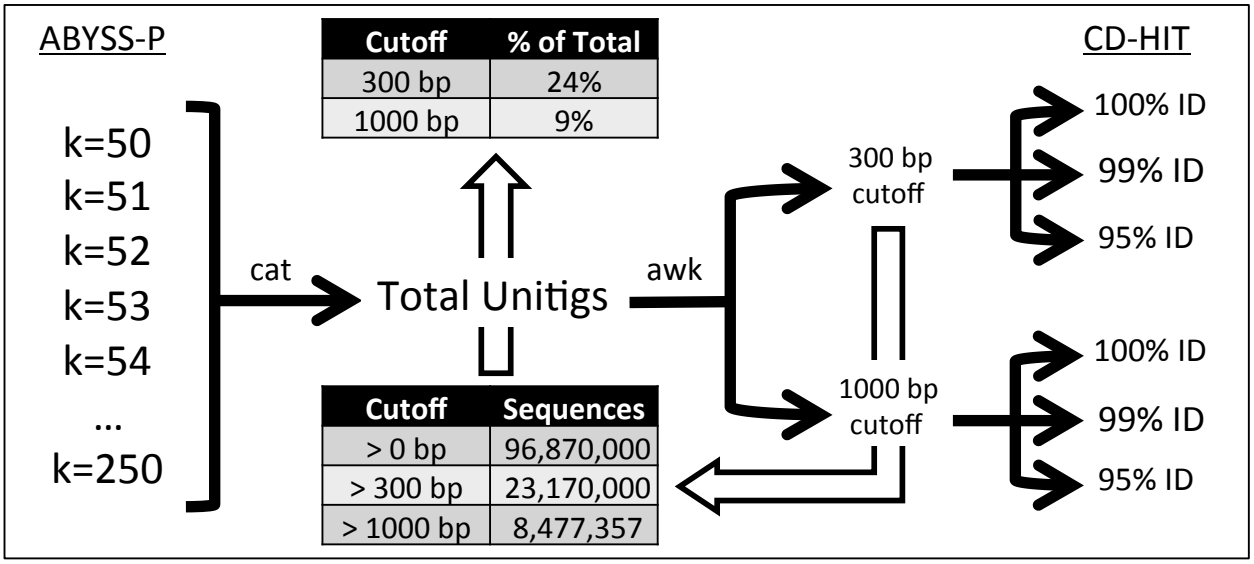


Figure 22. Overview of strategy to consolidate ABYSS contigs with CD-HIT. In order to consolidate redundant sequences, CD-HIT was tested with several different thresholds. Sequences were consolidated into clusters according to the indicated thresholds. A basic minimum length requirement of 1,000 bp removed 91% of the total assembled sequences. The command line utilities 'cat' and 'awk' were employed to join and filter the sequences respectively, prior to processing with CD-HIT.

Table 3. Summary of CD-HIT consolidation of ABYSS unitigs. This table summarizes the results of using CD-HIT to consolidate all of the ABYSS assemblies, with two different minimum length requirements. Even at 100% identity, the amount of redundancy eliminated among the set of sequences is substantial. Lowering the identity threshold further increases the amount of consolidation.

Size Cutoff (bp)	Percent Identity	Representative Sequences	Total (Mbp)
300	100	356,299	502.8
300	99	173,254	249.2
300	95	112,768	192.1
1000	100	147,856	390.8
1000	99	71,530	191.1
1000	95	56,091	158.9

Table 4. Summary of re-assembly of consolidated contigs. The SEQ1K 100 assembly was produced by consolidation with CD-HIT at a 100% identity threshold, and a minimum sequence length of 1,000 bp. The resulting assembly contained 147,856 contigs comprising 390.8 Mbp. The SEQ1K OLC assembly was produced by processing SEQ1K 100 with software from the ABySS toolkit (see Materials and Methods). The number of contigs was reduced by half and the number of bases was reduced to 215 Mbp. This demonstrates that the OLC strategy implemented with ABySS tools was able to further consolidate the sequences in the assembly.

Name	Contigs	L50	N50 (bp)	Max (bp)	Sum (Mbp)
SEQ1K 100	147,856	37,622	3,134	65,986	390.8
SEQ1K OLC	77,713	18,971	3,342	133,775	215.0

2.3.1.2 Assembling Illumina Data with DISCOVAR de novo

Since the library SXPX was created using a protocol that enabled compatibility with the specialist assembly program DISCOVAR *de novo*, the library could be assembled with this program. Unfortunately, the algorithms underlying DISCOVAR were never published and thus remain poorly understood except by those who designed them or have the time and expertise to dig into the very large code base. Unlike most other assembly programs, DISCOVAR has virtually no parameters that can be optimized, and is instead more of a "push button" assembler. One simply has to compile the code and then supply the right kind of sequencing library, and the program will return an assembly. Thus, operation of the program is relatively simple, though not well understood. The program does yield output that gives some insight into the processes occurring "under the hood" such as an error correction stage, a 60-mer DBG assembly stage, and a final 200-mer DBG assembly stage. When supplied with library SXPX, the total number of sequences in the final DISCOVAR assembly graph is 668,994 with a total of 321.3 Mbp (Table 5). This includes all 200-mers extracted from the reads. However, when excluding sequences less than 1,000 bp in length, there are 38,760 sequences and 156.0 Mbp in the assembly, with a N50 value of 5,621 bp and 52.98% GC content (Table 5). These numbers are remarkably better than any of the individual assemblies generated by ABYSS and also the consolidated assembly.

Using tools from ABySS, the final DBG from DISCOVAR was reconstructed by calculating all overlaps of 199 bp in the full assembly. The graph was then visualized using the program Bandage (*265*) for analysis of genome assembly graphs (Figure 6). This experiment revealed very interesting and complex structures in the assembly graph. The main feature of the graph is a massive "knot" which contains 75% of the nodes, 83% of the edges, and 45% of the total bases in the assembly graph. The rest of the graph consists largely of linear contigs and

unresolved sub-graphs. Visual inspection of the assembly graph illustrates the complexity associated with genome assembly and trying to determine sequences with a DBG-based approach.

2.3.1.3 Scaffolding and Gap Filling the DISCOVAR Assembly

Since the DISCOVAR assembly offered the best contiguity, it was used as a base to test scaffolding. At the time, only libraries SXPX, NGNB, and HOOW were available for this process. These libraries were aligned to the assembly with the BWA aligner using default settings and the alignments were coordinate sorted with SAMtools. The DISCOVAR contigs and the sorted alignments were then given as input to BESST to generate the scaffolds. The contigs were consolidated into 5,867 scaffolds with a N50 value of 129,421 bp and gaps accounted for 8.59% of the total bases (Table 6). Significantly, BESST generates scaffolds in passes using only one library at a time (*251*). By plotting the scaffold N50 of each pass against the fragment size of the libraries, a very strong linear correlation was observed between scaffold N50 and fragment size (Figure 7). This observation led to the conclusion that in order to obtain larger scaffolds, a new library was needed with a larger fragment size. This was in fact the inspiration to obtain the library LCHA, with a fragment size of approximately 15,000 bp, which would result in substantial improvements to scaffold quality.

With the availability of scaffolds containing large gaps, the next focus of testing was on gap filling. To achieve this, the PacBio data and the scaffolds were given as input to PBJelly. This program aligns the PacBio reads to the scaffolds and then analyzes the alignments to find reads that span or support gaps (*249*). Numerous challenges were encountered when running this program, as it was divided into several stages and the code base was not maintained in good alignment with its dependencies, primarily BLASR, the alignment program that it utilized. Despite

these issues, the program was successfully completed, and approximately 66% of gaps in the scaffolds were filled (Figure 8). However, by aligning the libraries to the scaffolds before and after gap filling, it becomes clear that there are misassembled gaps (Figure 9). The library HOOW in particular clearly shows the emergence of a small population of fragments with an unexpectedly large length (approximately 11 kb), indicating overfilled gaps. This is a known issue with PBJelly (*249*) and points to the importance of implementing error detection and correction after the gap filling stage of assembly.

Table 3. Statistics of DISCOVAR assembly. This table shows the contiguity statistics for the DISCOVAR assembly of the Illumina library SXPX. Excluding contigs shorter than 1 kb, the assembly captures approximately 93% of the estimated genome. However, the assembly is highly fragmented, and the contigs require further ordering and orientation (i.e. scaffolding).

Statistics	DISCOVAR v1
# contigs (>= 0 bp)	668,994
# contigs (>= 1 kbp)	38,760
# contigs (>= 10 kbp)	2,089
# contigs (>= 100 kbp)	36
# contigs (>= 1 Mbp)	2
Total length (>= 0 bp)	321,264,892
Total length (>= 1 kbp)	155,957,846
Total length (>= 10 kbp)	48,135,035
Total length (>= 100 kbp)	17,029,113
Total length (>= 1 Mbp)	3,056,987
Largest contig (bp)	1,739,873
GC (%)	52.98
N50 (bp)	5,621
L50	6,169
% N	0.51%

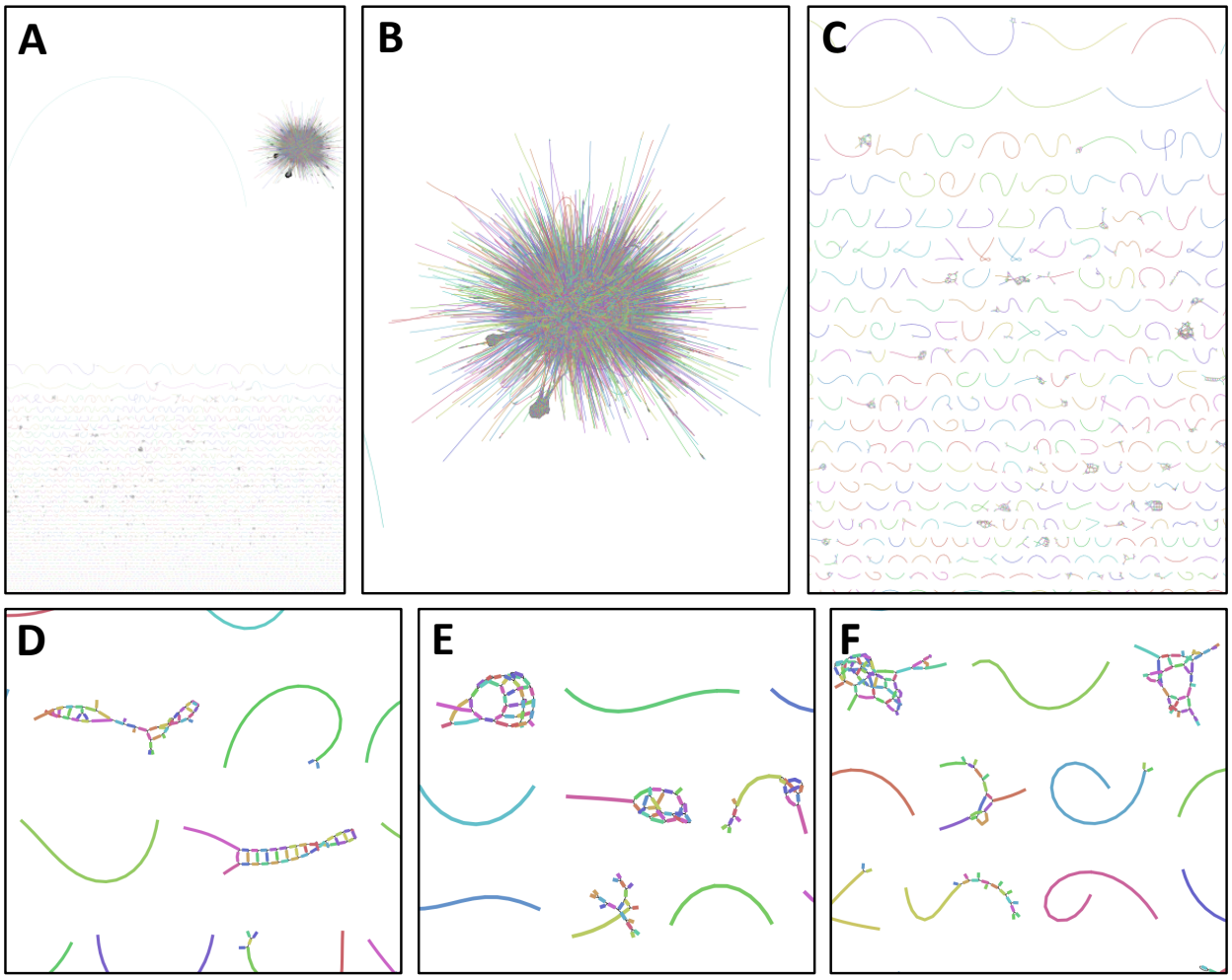


Figure 23. Visualization of DISCOVAR assembly graph with Bandage. The DISCOVAR program uses DBGs to generate the assembly. The final graph of DISCOVAR can be reconstructed from the complete set of contigs in the DISCOVAR output. The graph of all overlaps between contigs of k – 1 bp was reconstructed with 'abyss-overlap' from the ABySS toolkit. DISCOVAR uses a k-mer size of 200 for its final graph construction, and thus this value was used in the computation of overlaps. Panels A-F show snapshots of the total assembly graph that was reconstructed. It consists of a very large number of separate sub-graphs, variable in size. (A) Shows approximately a quarter of the total graph. (B) Zoomed in on an unusual feature, herein termed a "knot", which comprises 83% of the total edges in the graph. This feature likely results from highly repetitive DNA sequences. (C) Shows approximately 10% of the total graph, focused on the numerous linear sub-graphs that vary in size. (D-F) Show selections of the interesting graph structures reconstructed by DISCOVAR. However, it is also clear that many of the sequences are completely or mostly linear. These graphs highlight the complexity underlying genome assembly and the difficulty of resolving linear genomic sequences from DBGs.

Table 4. Statistics of scaffolded DISCOVAR assembly. This table shows the contiguity statistics of the scaffolds produced by BESST, using the DISCOVAR contigs greater than 1 kb in length as input. The four Illumina libraries were aligned against the contigs with HISAT2, and then the contigs and alignments were processed with BESST to yield scaffolds. The total number of sequences was reduced by 85%, with many contigs ordered and oriented into medium and large scaffolds. The total assembly size increased to 172.6 Mbp, only slightly above the estimated genome size.

	BESST v1
# contigs (>= 1 kb)	5,867
# contigs (>= 10 kb)	1,893
# contigs (>= 100 kb)	519
# contigs (>= 1 Mb)	8
Total length (>= 1 kb)	172,570,662
Total length (>= 10 kb)	162,975,113
Total length (>= 100 kb)	104,839,437
Total length (>= 1 Mb)	15,749,790
Largest contig	3,783,286
% GC	53.00%
N50 (bp)	129,421
L50	241
% N	8.59%

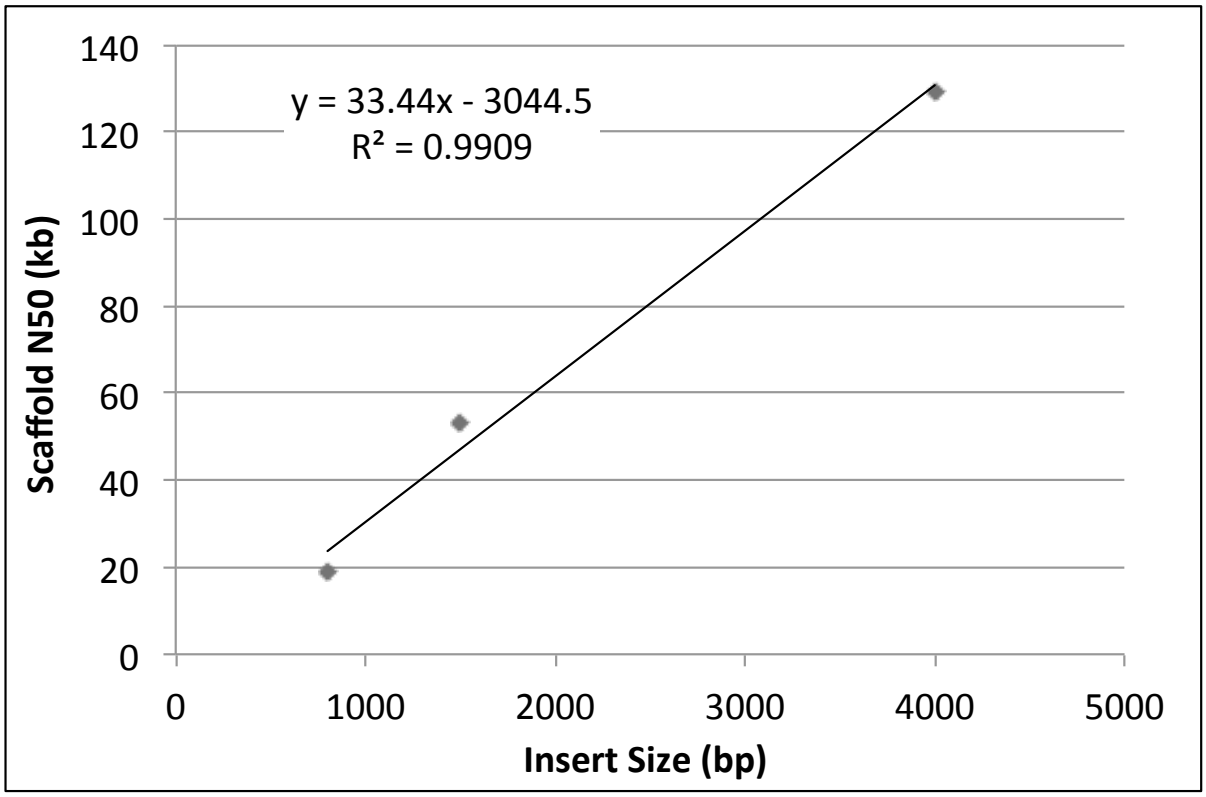


Figure 24. Comparison of library fragment size and scaffold N50. In the BESST scaffolding process, each library is processed individually, from smallest fragment size to largest fragment size. This graph shows the scaffold N50 after each pass of scaffolding. The fragment size very strongly correlates with the scaffold N50. This indicates that in order to obtain higher degrees of contiguity in the scaffolds, larger fragment sizes are needed to order and orient the contigs. This graph was produced before the Illumina library LCHA was constructed. In fact, it was this result that inspired the construction of the library LCHA. Using the above line equation, we estimated that a mate pair library with a 20 kb fragment size would yield a scaffold N50 of approximately 500 kb, giving a substantially higher degree of scaffold contiguity.

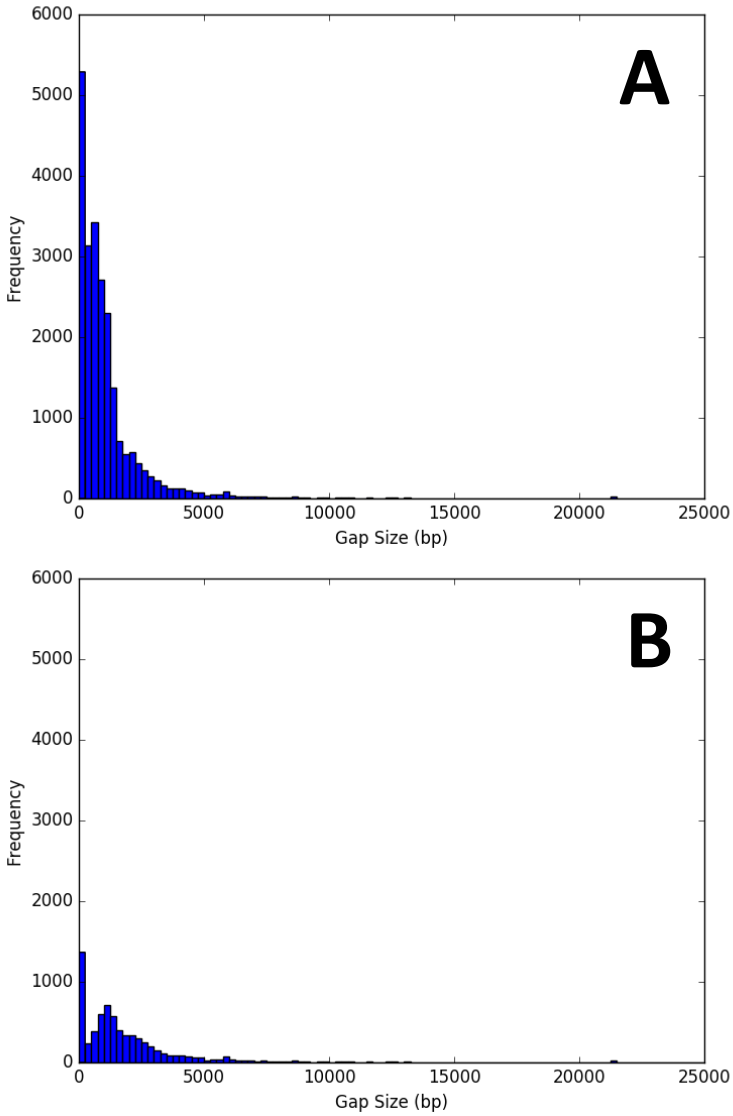


Figure 25. Closing gaps in the DISCOVAR scaffolds using PBJelly. There were a significant number of gaps in the assembly after scaffolding. In order to close these gaps, the PacBio data were utilized in conjunction with PBJelly. (A) Shows the initial distribution of gaps in the assembly. (B) Shows the distribution of gaps after application of PBJelly. These data demonstrate that PBJelly was effectively able to close a large number of gaps. However, the accuracy of these gap closures is not apparent from these data. Additionally, these data show that the larger gaps in the assembly are difficult to close.

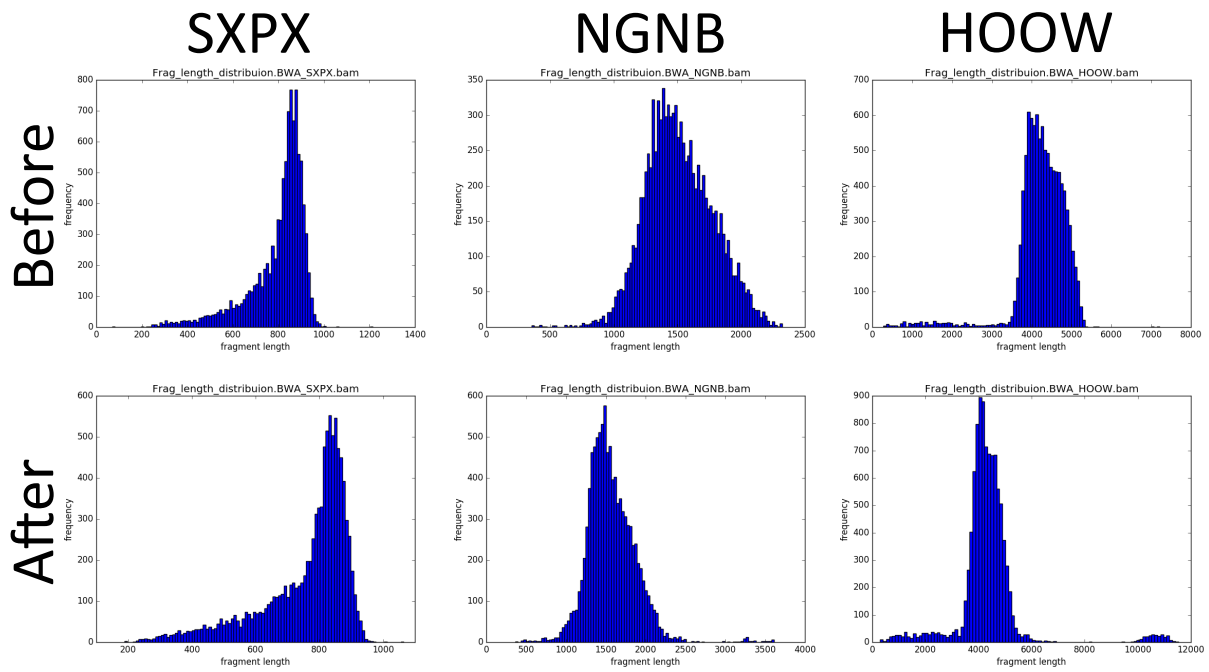


Figure 26. Analysis of library fragment sizes before and after gap filling. The fragment size distributions for each Illumina library were calculated on the initial assembly (scaffolds) and after gap-closing with PBJelly. While SXPX does not reveal much difference between the two states, the NGNB and HOOW libraries clearly show anamolously large fragments after gap-closing with PBJelly. These data indicate that PBJelly overfilled a number of gaps with probable mis-assembled sequences. This is a known issue with PBJelly and suggests that improvements are needed in the algorithm to avoid such mis-assemblies.

2.3.1.4 Assembling PacBio Data with FALCON and ABruijn

With the first round of PacBio sequencing data for *B. braunii* becoming available in May 2015, HudsonAlpha performed a preliminary assembly (i.e. FALCON v1) and delivered this assembly in November 2015. The result was an assembly consisting of 5,944 contigs with a N50 value of 45,930 bp and a total 152.2 Mbp of sequence, at 51.19% GC content (Table 7). Since this assembly did not offer substantial improvements in comparison to the Illumina-based assemblies that were under development, HudsonAlpha continued their efforts to assemble the PacBio data with FALCON. Subsequently, they delivered an updated assembly (i.e. FALCON v2) in June 2016. This time, the assembly consisted of 3,275 contigs with a N50 value of 175,683 bp and a total of 201.8 Mbp of sequence, at 51.29% GC content (Table 7). These statistics were comparable with the best Illumina assembly that had been constructed at the time. However, the assembly size of 201.8 Mbp substantially exceeded the estimated genome size of 166.2 Mbp, suggesting that further refinement of the assembly would be necessary.

Few algorithms were available in 2016 for assembling PacBio data and those that did exist were typically quite difficult to operate. Fortunately, in mid-2016, ABruijn was made available. This program was designed very well and was quite simple to operate, enabling easy experimentation. In October 2016, the program was downloaded and installed on the Ada supercomputer at Texas A&M University and used to assemble the PacBio data for *B. braunii*. In one of the initial tests of the program, two subsets of PacBio reads were created, with minimum read lengths of 10 kb and 6 kb, respectively giving approximate genome coverage of 40X and 100X. These two subsets were then given as input to ABruijn for assembly. The minimum 10 kb read set proved to have insufficient coverage and gave a very poor assembly, with only 81.8 Mbp of sequence assembled (Table 8). However, the minimum 6 kb read set yielded a very nice

assembly, consisting of 1,660 contigs with a N50 value of 121,321 bp and a total 165.5 Mbp of sequence, at 50.7% GC content (Table 8). Aside from genome coverage, the other major variable in ABruijn is the k-mer size, which is set to 15 by default. Different k-mer settings were tested, ranging from 13-16, with settings outside that range causing program failure. Adjusting the k-mer size to 14 resulted in a slight increase in assembly contiguity, with 1,624 scaffolds at a N50 value of 133,524 bp and a total of 172.4 Mbp of sequence, at 50.8% GC content (Table 9). Compared to the FALCON v2 assembly, there are half as many total contigs, but the assembly is generally less contiguous, with a slightly smaller N50 value. However, the total assembly size is closer to the estimated genome size.

Table 5. Statistics of FALCON assemblies from HudsonAlpha. These data show the assembly improvements made with iterations of FALCON at HudsonAlpha. While the contiguity improved substantially in version 2, the total assembly size became larger than the estimated genome size, for unknown reasons.

	FALCON_v1	FALCON_v2
# contigs (>= 0 bp)	5,944	3,275
# contigs (>= 1 kbp)	5,864	3,179
# contigs (>= 10 kbp)	3,642	2,245
# contigs (>= 100 kbp)	143	617
# contigs (>= 1 Mbp)	2	2
Total length (>= 0 bp)	152,229,687	201,800,654
Total length (>= 1 kbp)	152,177,795	201,742,451
Total length (>= 10 kbp)	140,521,785	196,646,609
Total length (>= 100 kbp)	27,269,629	140,423,151
Total length (>= 1 Mbp)	9,159,645	9,157,487
Largest contig	5,644,511	5,642,686
GC (%)	51.19%	51.29%
N50 (bp)	45,930	175,683
L50	895	320
# N's per 100 kbp	0	0

Table 6. Statistics of ABruijn assemblies at different minimum read lengths. These data show that the PacBio read set has insufficient coverage in reads longer than 10 kb to yield a high-quality assembly. Lowering the minimum read length threshold to 6 kb gave sufficient coverage to yield an assembly almost on par with those obtained from FALCON.

	Min_10kb	Min_6kb
# contigs (>= 1 kbp)	999	1,660
# contigs (>= 10 kbp)	999	1,660
# contigs (>= 100 kbp)	203	516
# contigs (>= 1 Mbp)	0	5
Total length (>= 1 kbp)	81,754,712	165,522,355
Total length (>= 10 kbp)	81,754,712	165,522,355
Total length (>= 100 kbp)	29,092,070	97,065,435
Total length (>= 1 Mbp)	0	8,614,004
Largest contig	525,988	3,510,543
GC (%)	50.15	50.7
N50 (bp)	83,777	121,321
L50	332	387
# N's per 100 kbp	0	0

Table 7. Statistics of ABruijn assemblies at different k-mer values. This experiment demonstrates the impact that the k-mer parameter of ABruijn has on the outcome of the algorithm. The k-mer size can be optimized to obtain better results, as shown by adjusting the k-mer size to 14, yielding the best N50 statistic and the higest number of contigs >= 100 kb.

	k = 13	k = 14	k = 15	k = 16
# contigs (>= 1 kb)	1,656	1,624	1,660	1,755
# contigs (>= 10 kb)	1,656	1,624	1,660	1,755
# contigs (>= 100 kb)	539	553	516	450
# contigs (>= 1 Mb)	2	4	5	1
Total length (>= 1 kb)	170,005,618	172,418,661	165,522,154	152,316,988
Total length (>= 10 kb)	170,005,618	172,418,661	165,522,154	152,316,988
Total length (>= 100 kb)	101,903,482	108,101,605	97,065,160	75,013,903
Total length (>= 1 Mb)	9,175,007	9,284,276	8,614,004	1,472,255
Largest contig	5,661,723	3,515,156	3,510,543	1,472,255
% GC	50.8%	50.8%	50.7%	50.6%
N50 (bp)	121,824	133,524	121,321	97,793
L50	387	364	387	462
# N's per 100 kbp	0	0	0	0

2.3.1.5 Comparing ABYSS, DISCOVAR, FALCON, and ABruijn Assemblies

Statistics describing contiguity, assembly size, and GC content do little to describe the actual sequence contents of each assembly and how they compare to each other. Looking at the coverage profiles of Illumina reads aligned against the assemblies gives some qualitative insight into their differences (Figure 10). Although these profiles are interesting, they do not give a quantitative comparison of the underlying sequence contents of the assemblies. A more effective measure of comparison is the Jaccard coefficient, which represents the cardinality of the intersection divided by the cardinality of the union between two sets (*266*). Perfectly identical sets have a Jaccard coefficient of one, while sets with no common elements have a Jaccard coefficient of zero. For any given value of k, the k-mer contents of an assembly can be considered a set, which completely describes the sequence contents of the assembly. Thus, the sequence similarity of two assemblies can be directly compared using the Jaccard coefficient of their k-mer contents. However, the results of this comparison are inherently dependent on the selection of k-mer size.

The total number of possible k-mers (P) is defined by $|A|^k$, where $|A|$ is the size of the alphabet, which is 4 in the case of DNA (i.e. adenine, thymine, guanine, and cytosine). As the value of k increases, P increases exponentially. In the case of *B. braunii*, with an estimated genome length (L) of 166.2 Mbp, with a k-mer size of 14, there could be a maximum of approximately 166.2 million unique k-mers (i.e. L – k + 1). When k equals 5, P equals 1,024 and it is thus expected to see each 5-mer in the *B.* braunii genome about 162 thousand times on average. Yet when k equals 14, P equals approximately 268.4 million. Thus a k-mer size of 14 gives a sufficiently large set of possible k-mers such that every 14-mer in the *B. braunii* genome could be unique. However, due to the presence of repetitive elements in the genome, many 14-mers are in fact found multiple times, and not every possible 14-mer is observed.

To examine the sequence similarity of the consolidated ABYSS, DISCOVAR, FALCON, and ABruijn assemblies, the Jaccard coefficient of the k-mer contents was calculated at four different values of k (i.e. 15, 20, 25, and 1,000) (Figure 11). To compare the differences between the Illumina and PacBio assemblies, the mean and standard deviation of the Jaccard coefficient was calculated using the three smaller values of k (15, 20, 25). The ABYSS and DISCOVAR assemblies have a mean Jaccard coefficient of about 0.85, while the ABruijn and FALCON assemblies have a mean Jaccard coefficient of about 0.75. When comparing either of the Illumina assemblies to either of the PacBio assemblies, the mean Jaccard coefficient is about 0.65. This suggests that although the majority of sequence is shared between the Illumina and PacBio assemblies, there are some unique sequences in each. One interpretation of this result is that combining the two sequence sets would give a more complete set of genomic sequence, which is not obtainable with a single sequencing technology. Thus future assembly algorithms should continue to explore the integration of multiple sequencing data types.

The comparison of the assemblies when k equals 1,000 is particularly interesting, because there is still a quite substantial amount of shared sequence (41-75%). Considering the unfathomably enormous number of possible 1,000-mers, and the potential for small differences emerging from the various assemblers, that the assemblies share any 1,000-mers at all seems impressive. This begged the question of how many unique, distinct, and total 1,000-mers were there in each of these assemblies? In other words, how many 1,000-mers were repeated? The Jellyfish k-mer counting software was utilized to count 1,000-mers in the different assemblies (Table 10). Remarkably, the maximum frequency of a 1,000-mer was 274 in the FALCON assembly, 199 in the ABruijn assembly, 11 in the DISCOVAR assembly, and 2 in the ABYSS assembly. Clearly, there are large differences in the repeat content of the Illumina assemblies and

the PacBio assemblies. To assess whether this phenomenon was unique to *B. braunii*, 1,000-mers were counted in the genomes of *V. carteri*, *C. reinhartdii*, and *A. thaliana* (Figure 12). While the FALCON assembly of the *B. braunii* genome shows the greatest degree of repetitive 1,000-mers, the other species also show a notable amount of repeated 1,000-mers. Based on these data, it seems possible that the high frequency of 1,000-mers in *B. braunii* is a real feature of its genome. Alternatively, it is possible that some of these sequences are actually artifacts of the PacBio data. Without further testing, such as PCR amplification from genomic DNA, it is difficult to conclude whether or not these sequences are true genomic elements.

In order to understand the distribution of these repeated 1,000-mers in the genomic sequences of *B. braunii*, the unique 1,000-mers were extracted from the ABruijn assembly and re-assembled with BCALM2 (Figure 13, Table 11). If the repeated 1,000-mers were confined to a small set of contigs, it would be expected that this re-assembly process would have a minimal impact on the assembly statistics. In contrast, if the repeated 1,000-mers were widely distributed throughout the genome, connecting unique regions, their removal would serve to fragment the assembly. The evidence aligns with the latter hypothesis, as re-assembly of the unique 1,000mers from the ABruijn assembly results in 7,464 contigs with a N50 value of 70,344 bp, compared to the original 1,660 contigs with a N50 value of 121,321 bp (Table 11). Re-assembly of the unique 1,000-mers resulted in a set of completely linear contigs (Figure 13). However, by allowing non-unique 1,000-mers into the assembly, the contigs are further fragmented (Figure 13). Moreover, when looking at the assembly graphs, a "knot" forms and grows with the maximum allowed k-mer frequency (Figure 13). This suggests that such assembly "knot" features, as observed in the DISCOVAR assembly graph, are the result of repetitive genomic content, and confound the assembly process.

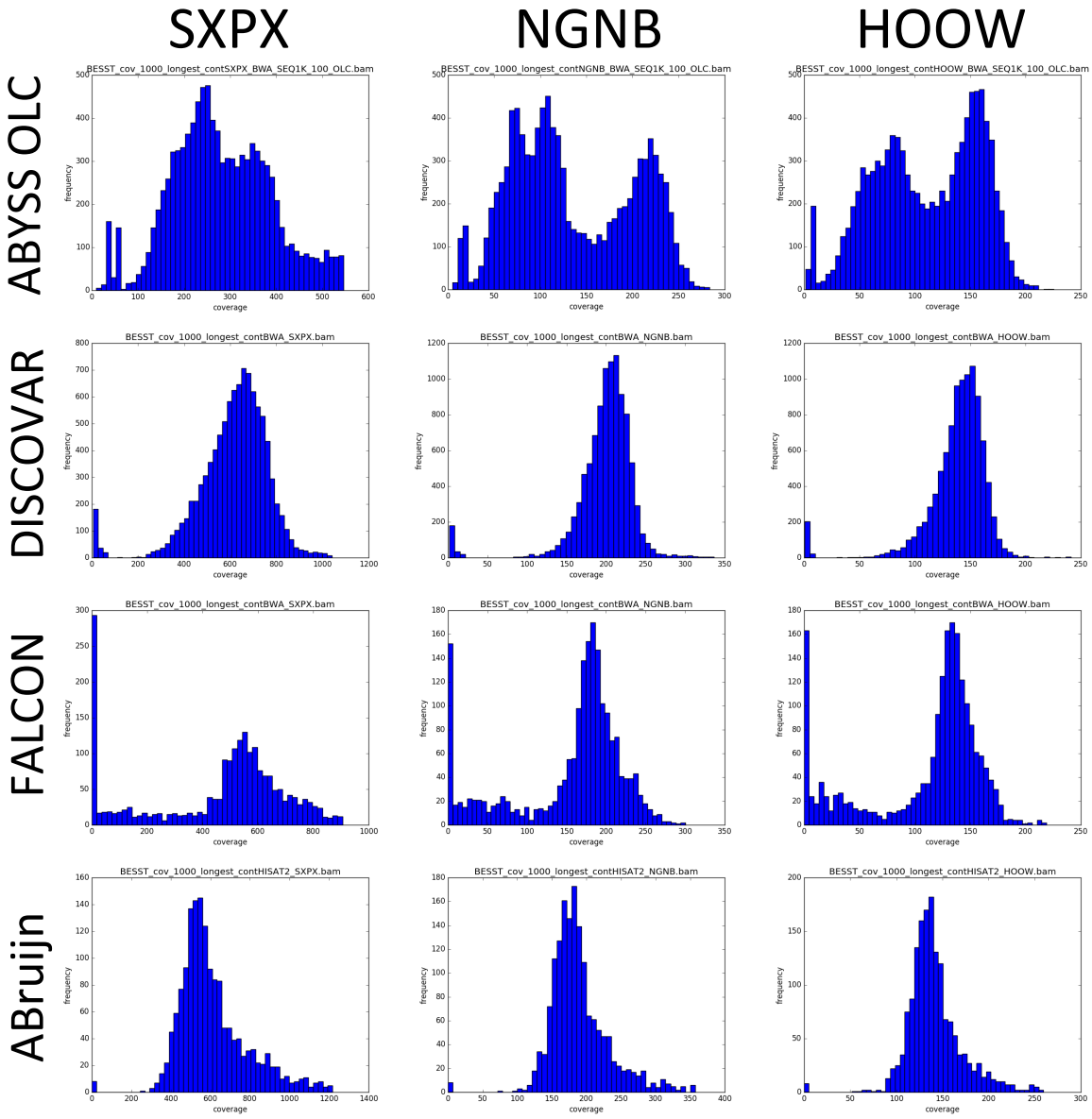


Figure 27. Illumina coverage profiles of different *B. braunii* genome assemblies. Each of the above assemblies show different coverage profiles after aligning the Illumina libraries against them with HISAT2. The ABYSS OLC assembly in particular shows a very distinct coverage profile. Whereas the other three assemblies (DISCOVAR, FALCON, and ABruijn) show fairly similar profiles. Although the DISCOVAR assembly was assembled from library SXPX, there are still sequences with no or very low coverage. The FALCON assembly has a large number of sequences that have no coverage in the Illumina datasets. However, the ABruijn assembly has few low-coverage sequences, indicating significant discrepancies in the assembly of PacBio data dependent on the method of assembly.

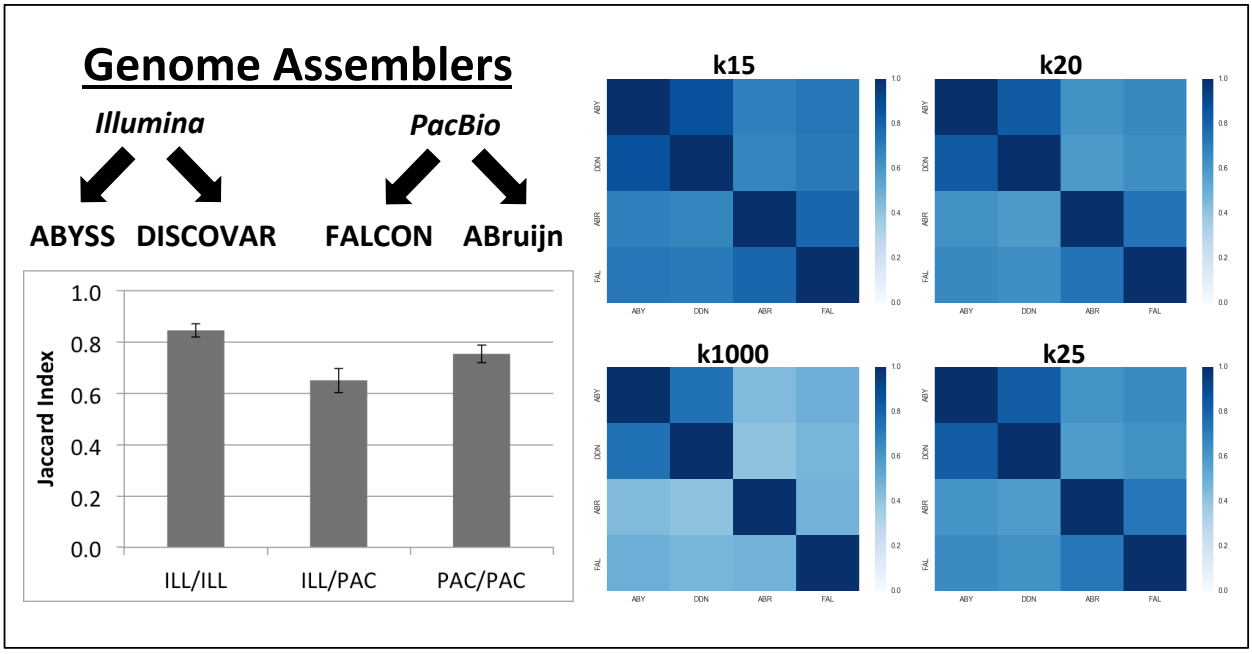


Figure 28. Comparison of sequence contents of different *B. braunii* genome assemblies. This experiment was intended to provide a more meaningful comparison of the sequences than the previous coverage-based analysis. By directly comparing the k-mer contents of each assembly, we can observe the absolute sequence similarity in terms of the Jaccard index. The results demonstrate that regardless of assembly method or sequencing data, a large fraction of core genomic sequences is recovered. However, there are clearly differences between both assembly methods and sequencing data. Essentially, there are sequences uniquely assembled from the Illumina and PacBio data. This indicates that a combination of both Illumina and PacBio data will yield a more complete model of the genome.

Table 8. Statistics of 1,000-mers in the different *B. braunii* assemblies. The number of possible DNA 1000-mers is sufficiently large to approach infinity. Yet a comparison of the four *B. braunii* genome assemblies revealed a substantial amount of shared 1,000-mers. This table presents a further analysis of the 1,000-mers found in each assembly. It reveals that the Illumina-based assemblies do not contain a large number of repeated 1,000-mers. Whereas the PacBio-based assemblies both have 1,000-mers that are highly repeated.

	Unique	Distinct	Total	Max Count
FALCON	168,082,682	169,865,645	175,868,948	274
ABYSS OLC	68,481,134	97,299,729	144,594,435	11
DISCOVAR	115,884,751	116,089,138	116,293,525	2
ABruijn	155,938,239	157,694,710	163,864,015	199

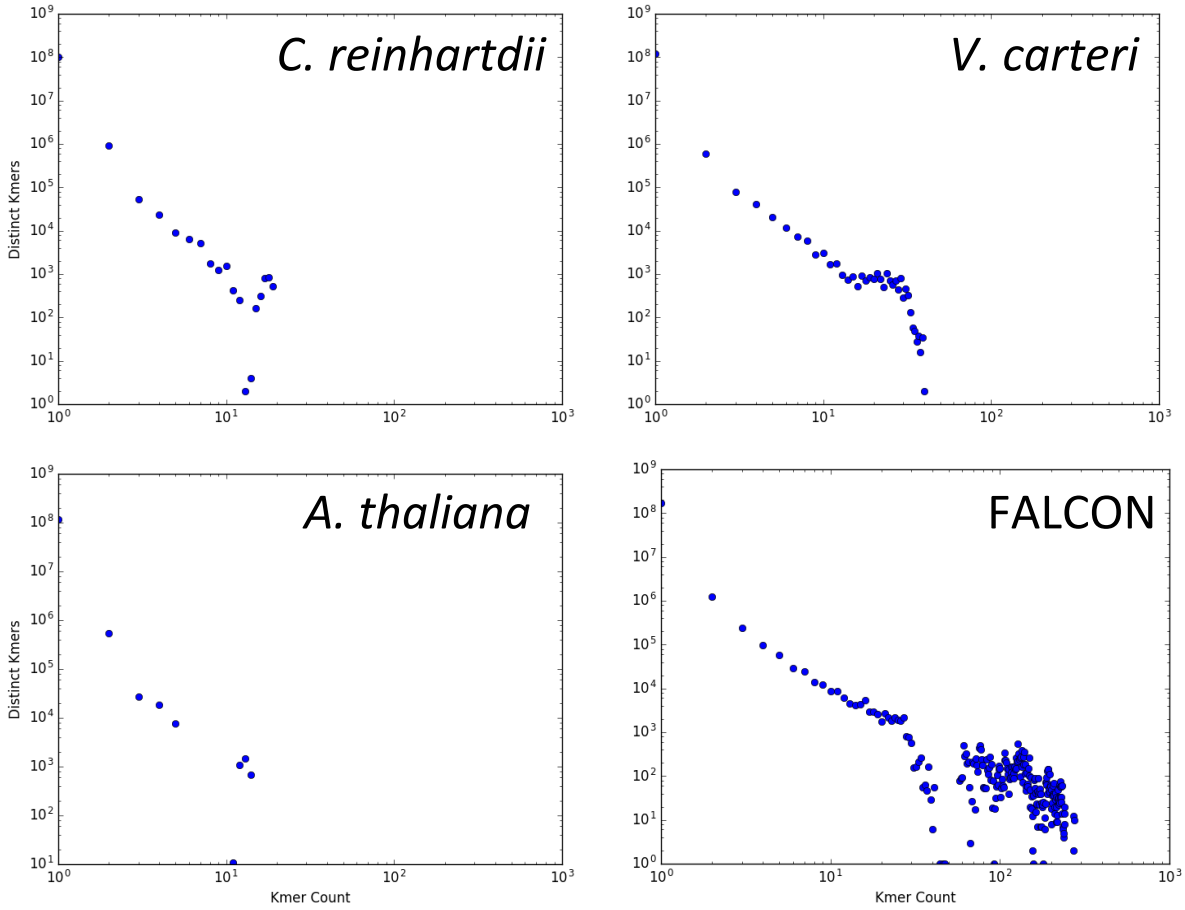


Figure 29. Genomic frequency distributions of 1,000-mers in various species. These data show in greater detail the frequency of occurrence for 1,000-mers found in the *B. braunii* version 1.0 genome (assembled with FALCON) and other species. While there are repeated 1,000-mers in the genome assemblies of other species, especially in *V. carteri*, the *B. braunii* genome assembly has by far the highest number of repeated 1,000-mers. Based on these data alone, it is difficult to determine whether the highly repetitive *B. braunii* 1,000-mers are true genomic sequences, or assembly artifacts from the PacBio data, or a combination of both.

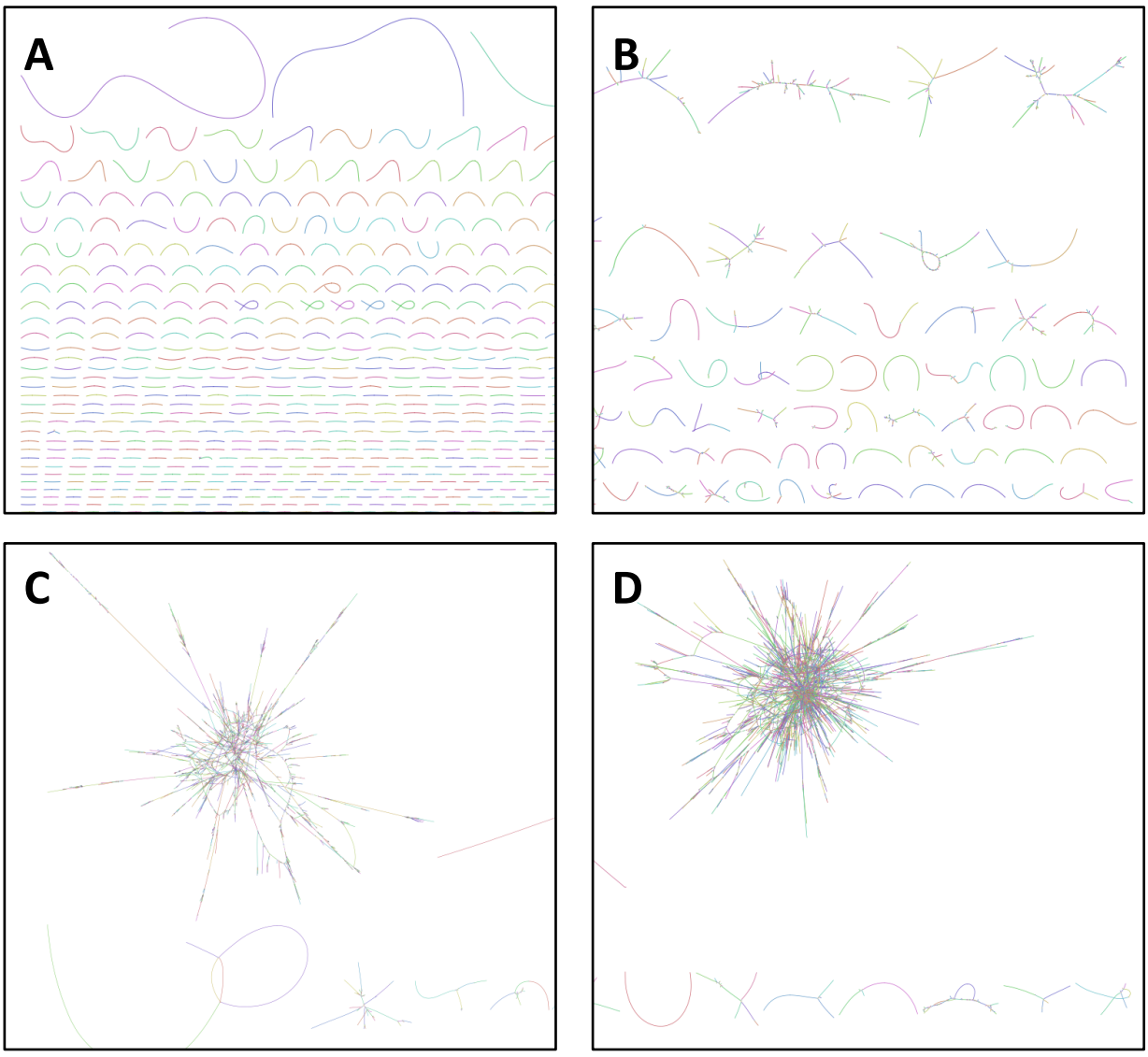


Figure 30. Re-assembly of ABruijn contigs with BCALM2 at variable maximum allowed 1,000-mer frequency. This experiment demonstrates the impact of recurring k-mers on de Bruijn graph structure. KAT was used to extract all 1,000-mers from contigs assembled by ABruijn with the PacBio data. Jellyfish was then used to sub-select 1,000-mers with a maximum count of 1 (A), 2 (B), 3 (C), and 4 (D). These 1,000-mer subsets were then re-assembled into de Bruijn graphs using BCALM2. (A) Shows that when all 1,000-mers are unique, the resulting contigs are perfectly linear. (B-D) Shows that repetitive k-mers increasingly confound contig assembly by adding edges to the de Bruijn graph, resulting in unresolvable graph structures.

Table 9. Statistics of ABruijn and BCALM2 assemblies. In further support of the fragmentation of de Bruijn graph assemblies by repetitive k-mers, this table shows the statistics of the BCALM2 re-assemblies of 1,000-mers from the ABruijn assembly. As 1,000-mers with higher counts are allowed into the assembly, the number of total contigs increases, and the N50 statistic decreases.

	ABruijn	BCALM2-1	BCALM2-2	BCALM2-3	BCALM2-4
# contigs (>= 1 kbp)	1,660	7,464	10,120	11,211	11,865
# contigs (>= 10 kbp)	1,660	2,610	2,612	2,613	2,613
# contigs (>= 100 kbp)	516	340	340	340	340
# contigs (>= 1 Mbp)	5	3	3	3	3
Total length (>= 1 kbp)	165,522,355	163,394,775	167,233,576	168,550,581	169,316,331
Total length (>= 10 kbp)	165,522,355	149,064,588	149,117,103	149,126,204	149,125,474
Total length (>= 100 kbp)	97,065,435	55,634,067	55,635,513	55,635,513	55,635,513
Total length (>= 1 Mbp)	8,614,004	3,857,775	3,857,775	3,857,775	3,857,775
Largest contig	3,510,543	1,675,982	1,675,982	1,675,982	1,675,982
GC (%)	50.7%	50.72%	50.68%	50.67%	50.66%
N50	121,321	70,344	68,165	67,428	67,085
L50	387	655	683	693	699

2.3.2 Building the Version 2.0 Genome of B. braunii

With the lessons learned from the initial experimentation in genome assembly, and the recent emergence of new sequencing data for *B. braunii*, efforts were made to finalize a draft genome for publication. The following sections describe the iterative efforts undertaken thus far to achieve this goal, resulting first in the "Version 1.0" and then the "Version 2.0" genome assemblies. The details of the assembly methods are presented and discussed, as well as further lessons learned and opportunities for future improvements. In summary, the pipeline consists of separately assembling the Illumina and PacBio data with purpose-built assemblers, merging the two assemblies, scaffold, gap-filling, polishing, quality filtering, and re-scaffolding to yield the final assembly for the "Version 2.0" release (Figure 14).

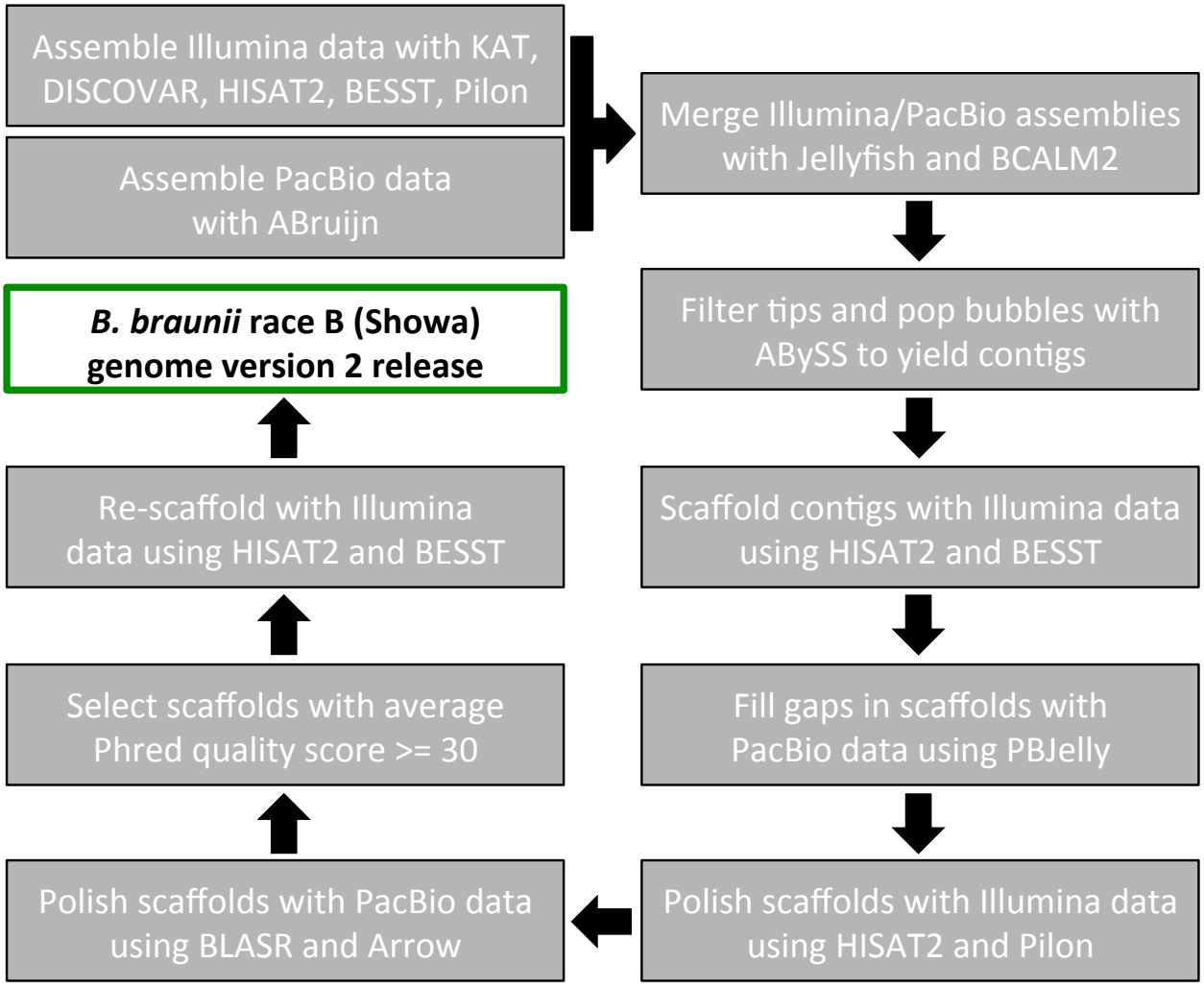


Figure 31. Summary of assembly pipeline for *B. braunii* genome Version 2.0. This figure presents an overview of the processes that were utilized to assemble the sequencing data for the *B. braunii* genome. The goal in developing this pipeline was to utilize existing tools to integrate the Illumina and PacBio data. By combining the two types of sequencing data, we aimed to obtain a more complete and higher quality assembly than before.

2.3.2.1 Assembling the Illumina and PacBio Data

The Illumina data was assembled with essentially the same tools that had been utilized in the earlier assembly experiments. One significant difference in this version was the inclusion of a filtration step for library SXPX prior to assembly with DISCOVAR. Utilizing a program suite called KAT (K-mer Analysis Toolkit) (*267*), 200-mers were counted in the library SXPX. There was a large amount of very highly repetitive 200-mers in the library, indicative of repeat content (Figure 15). Due to the observation that repetitive k-mers confound DBG assemblers, it was hypothesized that removal of this repeat content could improve the assembly. In order to filter out repeat content, KAT was used to select reads that did not contain any 200-mers with a total frequency greater than 500 counts in the library. The filtration process reduced the number of read pairs in the library from 249.5 million to 195.4 million. Although the total number of read pairs was reduced to 78.3% of the original, 91.5% of the distinct 200-mers were retained through the filtration process (Table 12). Thus, the repeat content of the library was substantially reduced with only minimal loss of distinctive sequence information.

After filtration, the library was given as input to DISCOVAR *de novo* to assemble contigs. Scaffolds were generated by selecting contigs greater than 1 kb, aligning the Illumina libraries against them with HISAT2, and giving this information as input to BESST. Finally, Pilon was used to correct errors in the assembly by aligning the Illumina libraries against the scaffolds with HISAT2 and giving this information as input. The final Illumina assembly consisted of 5,367 sequences with a N50 value of 415,985 bp, for a total of 175.0 Mbp at 53.1% GC content and 13.7% gap content (Table 13). The PacBio data were assembled with ABruijn as described previously, using reads greater than 6 kb in length and a k-mer size of 14. The final PacBio assembly consisted of 1,624 sequences with a N50 value of 133,524 bp, for a total of 172.4 Mbp

at 50.8% GC content and no gaps (Table 14). Although the Illumina assembly is more contiguous, it also contains many small fragments less than 10 kb in length. When excluding these small fragments, the Illumina assembly consists of 858 sequences for a total of 164.2 Mbp of sequence, which is just underneath the estimated genome size.

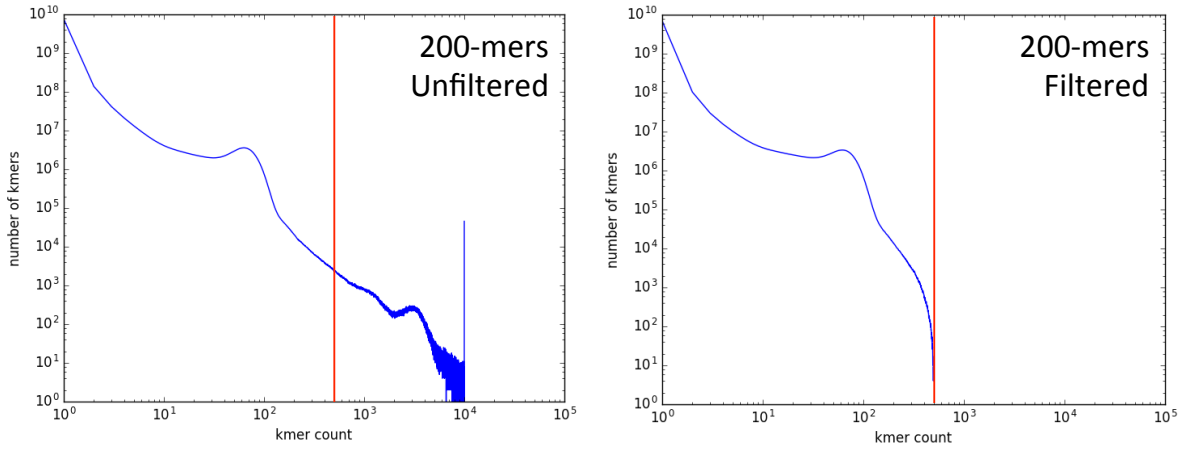


Figure 32. Distribution of 200-mers before and after filtering library SXPX. This data shows the 200-mer frequencies in the Illumina library SXPX as counted by Jellyfish, before and after filtering the library with KAT. The goal of this filtration experiment was to remove highly repetitive sequences from the library that would confound the assembler. The red lines indicate the frequency threshold of 500 counts.

Table 10. Summary of 200-mer filtering results for library SXPX. This table demonstrates that although a large number of reads were removed from the library by the filtration process, the vast majority of sequence information was retained.

	Read Pairs	Fragment Coverage	Sequence Coverage	Distinct 200-mers
Unfiltered	249,536,701	1,203	755	8,063,200,696
Filtered	195,455,926	942	591	7,378,101,607
Retained	78.3%	78.3%	78.3%	91.5%

Table 11. Summary of Illumina assembly statistics. The contigs were highly consolidated by the scaffolding process, resulting in an assembly of moderate quality. The polishing process had almost no impact on the contiguity and gap content of the assembly.

	DISCOVAR	BESST	Pilon
# contigs (>= 1 kbp)	34,727	5,370	5,367
# contigs (>= 10 kbp)	2,203	859	858
# contigs (>= 100 kbp)	35	393	394
# contigs (>= 1 Mbp)	2	15	15
Total length (>= 1 kbp)	153,041,564	174,524,300	174,985,090
Total length (>= 10 kbp)	49,745,997	163,713,670	164,176,541
Total length (>= 100 kbp)	17,000,021	148,848,424	149,406,841
Total length (>= 1 Mbp)	3,056,987	30,475,062	30,573,245
Largest contig	1,739,873	7,819,235	7,827,854
GC (%)	53%	53.10%	53.11%
N50	6,054	415,769	415,985
L50	5,721	112	112
# N's per 100 kbp	50	14,396	13,690

Table 12. Summary of Illumina and PacBio assembly statistics. Overall, the PacBio assembly is less contiguous than the Illumina assembly, as shown by the N50 statistics. However, the Illumina assembly also has a large number of small fragments (< 10kb) and a significant amount of gaps in the scaffolds (13.69%).

	Illumina	PacBio
# contigs (>= 1 kb)	5,367	1,624
# contigs (>= 10 kb)	858	1,624
# contigs (>= 100 kb)	394	553
# contigs (>= 1 Mb)	15	4
Total length (>= 1 kb)	174,985,090	172,418,661
Total length (>= 10 kb)	164,176,541	172,418,661
Total length (>= 100 kb)	149,406,841	108,101,605
Total length (>= 1 Mb)	30,573,245	9,284,276
Largest contig	7,827,854	3,515,156
% GC	53.1%	50.8%
N50 (bp)	415,985	133,524
L50	112	364
# N's per 100 kbp	13,690	0

2.3.2.2 Merging the Illumina and PacBio Assemblies

To merge the Illumina and PacBio assemblies, a DBG-based approach was utilized. Unique k-mers were extracted from each assembly, combined into a single set of k-mers, and then re-assembled. In order to maximally preserve unique content, a large k-mer size of 1,000 was selected. In the Illumina assembly, there were 116.9 million unique 1,000-mers, while in the PacBio assembly there were 159.0 million unique 1,000-mers (Table 15). After combining the two sets of unique 1,000-mers, there were 251.9 million distinct k-mers in the set. Thus between the two assemblies, there were approximately 24 million shared 1,000-mers (Jaccard coefficient of 0.095). The set of combined 1,000-mers was given as input to BCALM2 with the k-mer size set to 1,000 in order to construct a DBG. Manual inspection of the raw assembly graph from BCALM2 revealed that graph processing was needed in order to obtain contigs (Figure 16). Thus, programs from the ABySS toolkit were utilized to filter tips and pop bubbles. Finally, contigs less than 2 kb in length were discarded. The final set of contigs consisted of 13,395 sequences with a N50 value of 34,549 bp, for a total of 189.5 Mbp at 52.9% GC content (Table 16).

This work represents the first known DBG-based approach to merging assemblies. While it does not preserve the higher-order layout of the input assemblies, it completely merges the sequences, essentially folding them together like two decks of cards. Much more work could be done testing and optimizing such a DBG-based approach to assembly consolidation. How many assemblies can be simultaneously combined? What is the optimal k-mer size for merging assemblies? Could any pre-processing or post-processing steps be implemented to improve DBG construction and resolution? Exploring the answers to these questions and others could yield valuable insights into assembly consolidation processes. This domain of research is very much necessary for broadly improving the quality of complex eukaryotic genome assemblies.

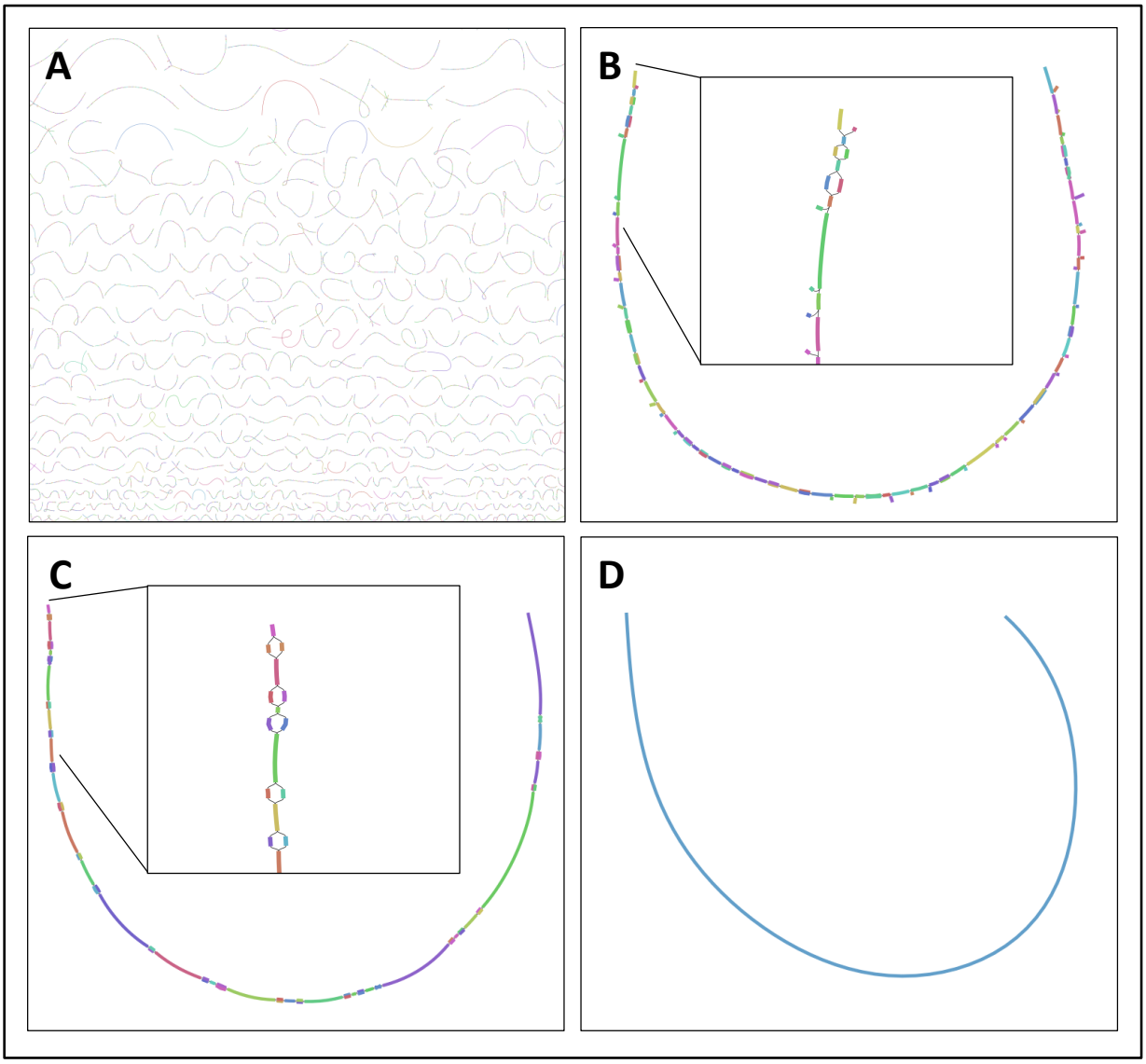


Figure 33. Visualization of assembly graph during tip filtration and bubble popping. These data show the various stages in assembly graph processing. (A) Shows a small selection of all the subgraphs in the total assembly graph. (B) Shows an example subgraph with tips and bubbles. (C) Shows a graph with only bubbles remaining. (D) Shows a complete, linear contig that results after filtering tips and popping bubbles.

Table 13. Summary of 1,000-mer merging. This table shows the number of 1,000-mers in the Illumina and PacBio assemblies. After the two sets of 1,000-mers were merged, the combined set contained nearly 252 million distinct 1,000-mers. These data show that the Jaccard index between the two assemblies is roughly 0.1, indicating little overlap between their 1,000-mers.

1000-mers	**Illumina**	**PacBio**	**Merged**
Unique	116,929,736	158,969,124	227,864,096
Distinct	116,929,736	158,969,124	251,881,478
Total	116,929,736	158,969,124	275,898,860

Table 14. Summary of assembly statistics during tip filtration and bubble popping. After re-assembling the combined 1,000-mers into a de Bruijn graph, there was a large amount of sequence in the assembly. Much of this excess sequence was due to the presence of tips and bubbles, and was removed by filtering out these features.

	Sequences	L50	Min (bp)	N50 (bp)	Max (bp)	Sum (Mbp)
Unitigs	120,966	25,559	1,000	3,281	1,246,125	349.0
Tipless	59,408	6,988	1,000	7,507	1,246,125	260.2
Popped	23,011	1,606	1,000	31,095	1,246,125	203.2
Contigs	13,395	1,383	2,000	34,549	1,246,125	189.5

2.3.2.3 Scaffolding, Gap Filling, and Polishing

The hybrid contigs were scaffolded with BESST by aligning the Illumina libraries against them with HISAT2 and then giving this information as input. The resulting scaffolds consisted of 4,455 sequences with a N50 value of 485,136 bp, for a total of 196.2 Mbp at 52.67% GC content and 7.5% gap content (Table 17). In order to detect and dismantle misassemblies in the scaffolds, a tool called REAPR was employed (*268*). The REAPR program takes as input an alignment of paired end reads against sequences and analyzes the fragment coverage distribution (FCD). Errors in the assembly are essentially defined as regions in which the observed FCD deviates too much from the expected FCD (see paper for complete details). The output of REAPR is a set of sequences that have been "broken" at regions defined as errors. After application of REAPR, the assembly consisted of 4,739 sequences with a N50 value of 381,540 bp, with essentially no change in total assembly size, GC content, or gap content.

Gaps in the assembly were filled with the PacBio data using PBJelly, followed by polishing with the Illumina data using Pilon, and then with the PacBio data using Arrow (the successor to Quiver, available at https://github.com/PacificBiosciences/GenomicConsensus but currently unpublished). The application of PBJelly reduced the gap content in the assembly from 7.5% to 5.4%, with the two polishing steps closing a few more gaps, for a final of 5.3% gap content (Figure 17). After gap filling and prior to polishing, REAPR was applied to detect and break errors introduced by PBJelly. The final polished scaffolds consisted of 4,866 sequences with a N50 value of 273,032 bp, for a total 197.2 Mbp at 52.74% GC content and 5.3% gap content (Table 18). One of the major advantages of polishing with the PacBio data using Arrow is that the output includes a FASTQ file of the assembly, giving quality values for every base. This enables the calculation of average base quality scores per scaffold. Interestingly, the polishing with Arrow revealed that a

substantial fraction of sequences in the assembly have very low average base quality (Figure 18). Furthermore, some of these low-quality sequences are in fact among the longest sequences in the assembly. It is possible that these are non-algal sequences, such as bacteria, which are co-cultivated with the algae. The presence of bacteria in the algal cultures means that bacterial DNA may have been isolated along with the algal DNA in the process of sequencing library preparation. However, since the scope of this work is to assemble the genome of *B. braunii*, the origin of these large, low-quality sequences was not further investigated.

Table 15. Statistics of *B. braunii* genome Version 2.0 through scaffolding. These data show that the scaffolding process greatly consolidated the contigs into larger fragments. The number of bases in scaffolds > 100 kbp closely approximates the estimated genome size (166 Mbp). Some of the larger pieces are broken apart when errant linkages are detected by REAPR.

	Contigs	Scaffolds	Broken
# contigs (>= 1 kbp)	13,395	4,455	4,739
# contigs (>= 10 kbp)	4,424	939	1,135
# contigs (>= 100 kbp)	131	385	465
# contigs (>= 1 Mbp)	3	27	15
Total length (>= 1 kbp)	189,480,694	196,239,172	196,330,972
Total length (>= 10 kbp)	153,699,013	184,669,233	184,440,427
Total length (>= 100 kbp)	28,562,334	168,454,348	163,306,896
Total length (>= 1 Mbp)	3,546,982	44,638,172	28,577,098
Largest contig	1,246,125	4,640,634	4,640,634
GC (%)	52.87%	52.67%	52.67%
N50 (bp)	34,549	485,136	381,540
L50	1,383	105	139
# N's per 100 kbp	405	7,535	7,581

Table 16. Statistics of *B. braunii* genome Version 2.0 through gap filling and polishing. The gap filling process with PBJelly reduces the gap content by 2.1%. However, it also introduces a substantial number of errors into the assembly, which is further fragmented by another round of REAPR. The rounds of polishing with Illumina and PacBio data have little impact on assembly contiguity.

	Gap-filled	Broken	Illumina Polish	PacBio Polish
# contigs (>= 1 kbp)	4,451	4,866	4,866	4,866
# contigs (>= 10 kbp)	1,136	1,418	1,416	1,420
# contigs (>= 100 kbp)	462	527	527	527
# contigs (>= 1 Mbp)	15	9	9	9
Total length (>= 1 kbp)	196,491,430	196,434,805	196,579,807	197,278,782
Total length (>= 10 kbp)	185,495,392	184,793,256	184,931,082	185,619,742
Total length (>= 100 kbp)	164,529,231	154,663,269	154,838,483	155,387,241
Total length (>= 1 Mbp)	28,576,937	20,355,213	20,353,705	20,375,393
Largest contig	4,640,634	4,640,634	4,640,229	4,640,271
GC (%)	52.65%	52.65%	52.66%	52.74%
N50	393,324	270,817	272,332	273,032
L50	136	187	186	186
# N's per 100 kbp	5,442	5,416	5,356	5,279

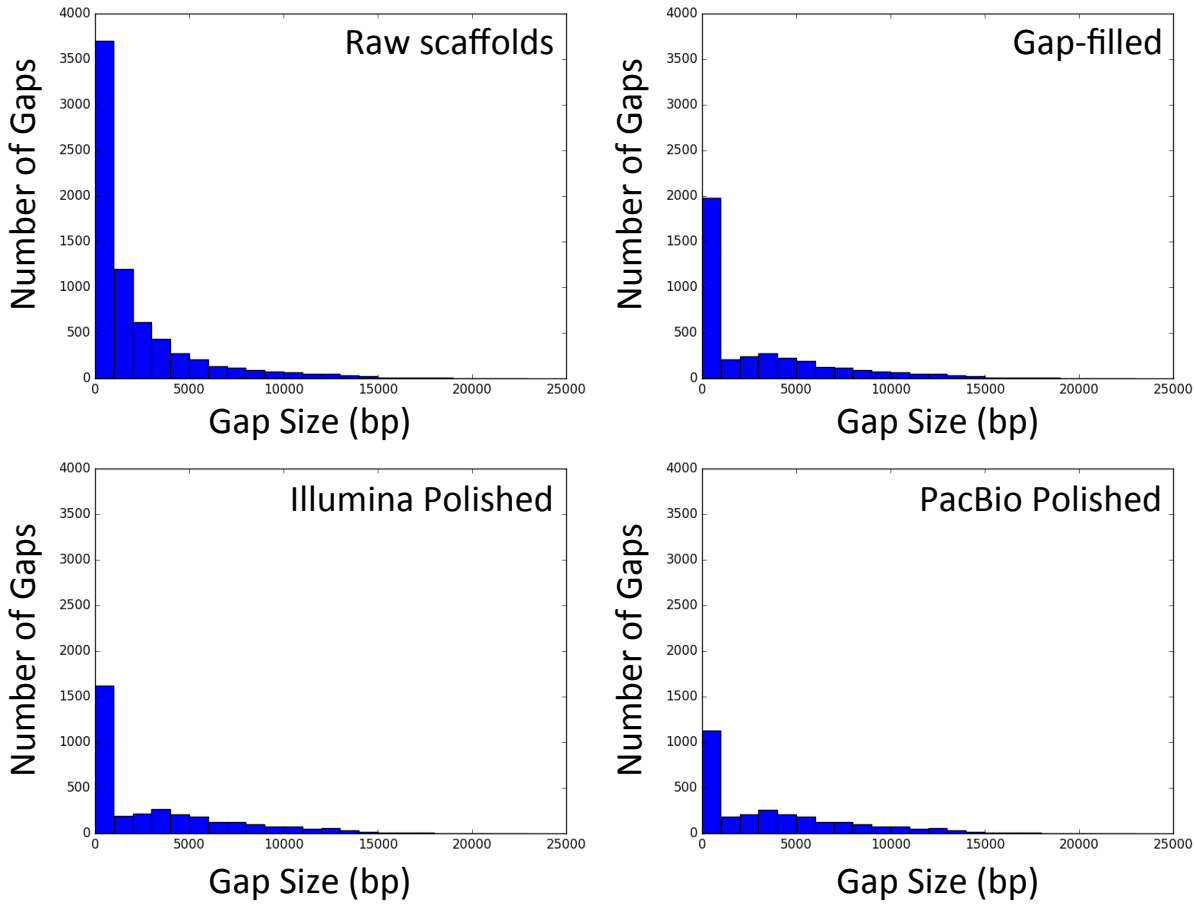


Figure 34. Gap size distribution throughout gap filling and polishing. The total number of gaps was reduced from 7,104 in the raw scaffolds to 2,859 after gap filling and double polishing. These data show that a large number of small gaps were closed, but many of the larger gaps remain in the scaffolds, for a total 5.3% of the assembly.

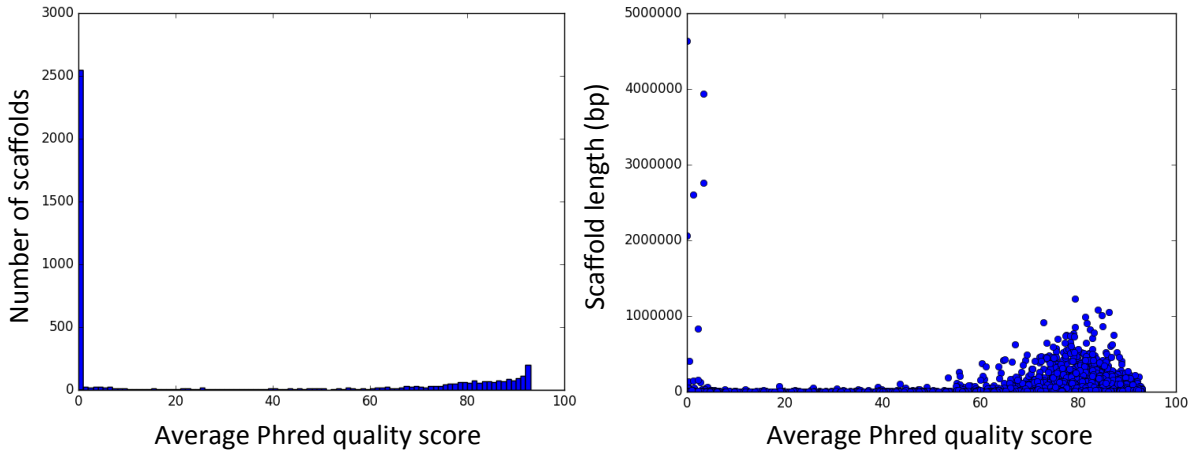


Figure 35. Summary of average scaffold quality scores and lengths after polishing. The base quality information from PacBio polishing enables the calculation of average quality scores for each scaffold. These data show a clear separation of low- and high-quality scaffolds, with several of the longest scaffolds, and many of the smallest scaffolds, having very low scores.

2.3.2.4 Quality Filtering and Re-scaffolding

The availability of quality scores per scaffold enables the separation of the total set of scaffolds into sets of high-quality and low-quality sequences. A Phred score of 30 was chosen as the threshold, meaning that the average quality is greater than 99.9% confidence per base called. Using this threshold, the high-quality scaffolds consisted of 1,972 sequences with a N50 value of 260,708 bp, for a total of 169.2 Mbp at 50.81% GC content and 5.99% gap content (Table 19). Separating the high-quality and low-quality scaffolds revealed that they have distinctive GC content and fragment coverage signatures (Figure 19). Furthermore, analysis of the Illumina sequence coverage profiles throughout the assembly pipeline reveals major changes in the fundamental sequence characteristics (Figure 20). This data clearly illustrates that even after the initial assembly; sequence properties can change substantially, depending on how the sequences are processed. This highlights the need for a deeper fundamental understanding of assembly.

Lastly, the high-quality scaffolds were re-scaffolded with BESST by aligning all of the Illumina libraries against them with HISAT2 and giving this information as input. HudsonAlpha evaluated scaffold quality by aligning the library LCHA and calculating fragment coverage per base. The assembly was broken at regions where there were no supporting fragments. The final "Version 2.0" scaffolds consisted of 983 sequences with a N50 value of 565,620 bp, for a total of 170.2 Mbp at 50.82% GC content and 7.3% gap content (Table 20). In terms of contiguity and proximity to the expected genome size, the "Version 2.0" assembly has substantial improvements compared to the "Version 1.0" assembly (Table 21). However, contiguity does not provide insight into deeper assembly quality such as recovery of gene content. Thus, further evaluation is needed, built on gene predictions and functional annotations. Such analysis will be described in the following section.

Table 17. Assembly statistics after quality filtration. There are substantially more low-quality scaffolds than high-quality scaffolds. Most of the scaffolds >= 100 kb are in the high-quality category, but 5 of the largest scaffolds are low-quality. The other low-quality sequences are almost entirely small fragments < 10 kb.

	High-Quality	Low-Quality
# contigs (>= 1 kb)	1,972	2,894
# contigs (>= 10 kb)	1,326	94
# contigs (>= 100 kb)	516	11
# contigs (>= 1 Mb)	4	5
Total length (>= 1 kb)	169,154,875	28,123,907
Total length (>= 10 kb)	165,918,982	19,700,760
Total length (>= 100 kb)	137,580,125	17,807,116
Total length (>= 1 Mb)	4,374,823	16,000,570
Largest contig	1,228,030	4,640,271
% GC	50.81%	63.76%
N50 (bp)	260,708	2,062,008
L50	191	5
% N	5.99%	0.99%

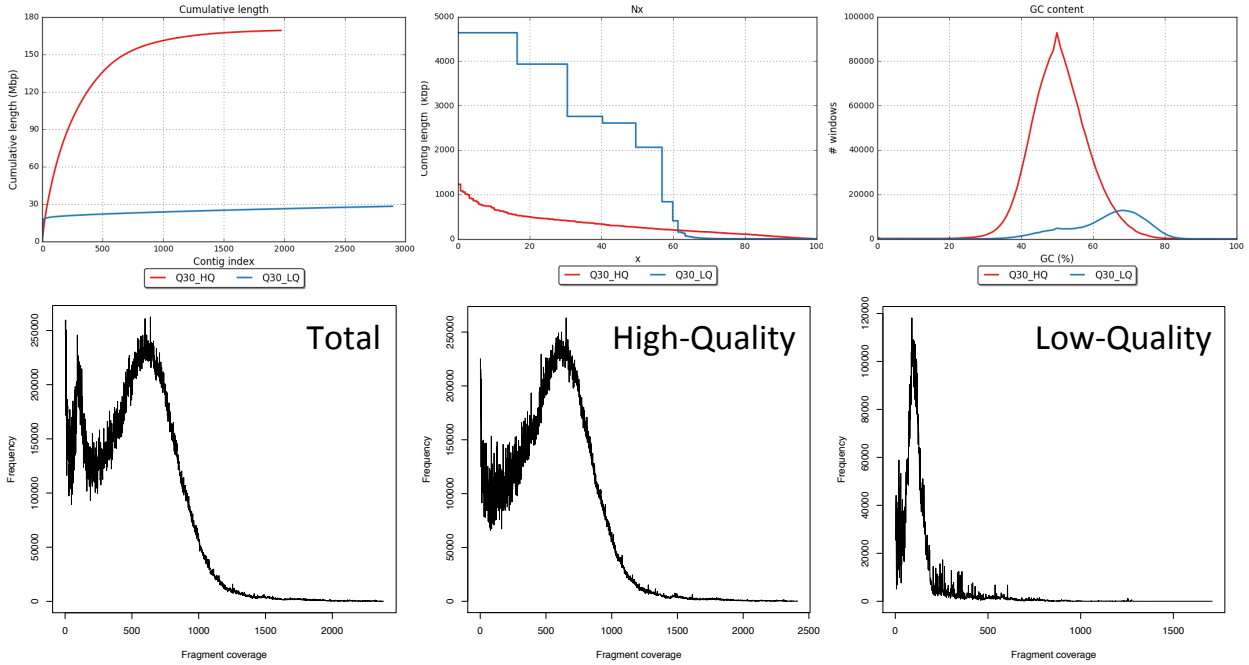


Figure 36. Summary of properties of high-quality and low-quality sequences. The low-quality sequences have remarkably different fragment coverage profiles and GC contents. The data suggest that these could be entirely separate genomes that were partly or entirely co-assembled with the algal genome. It is very important to separate these contaminating sequences prior to any downstream analyses.

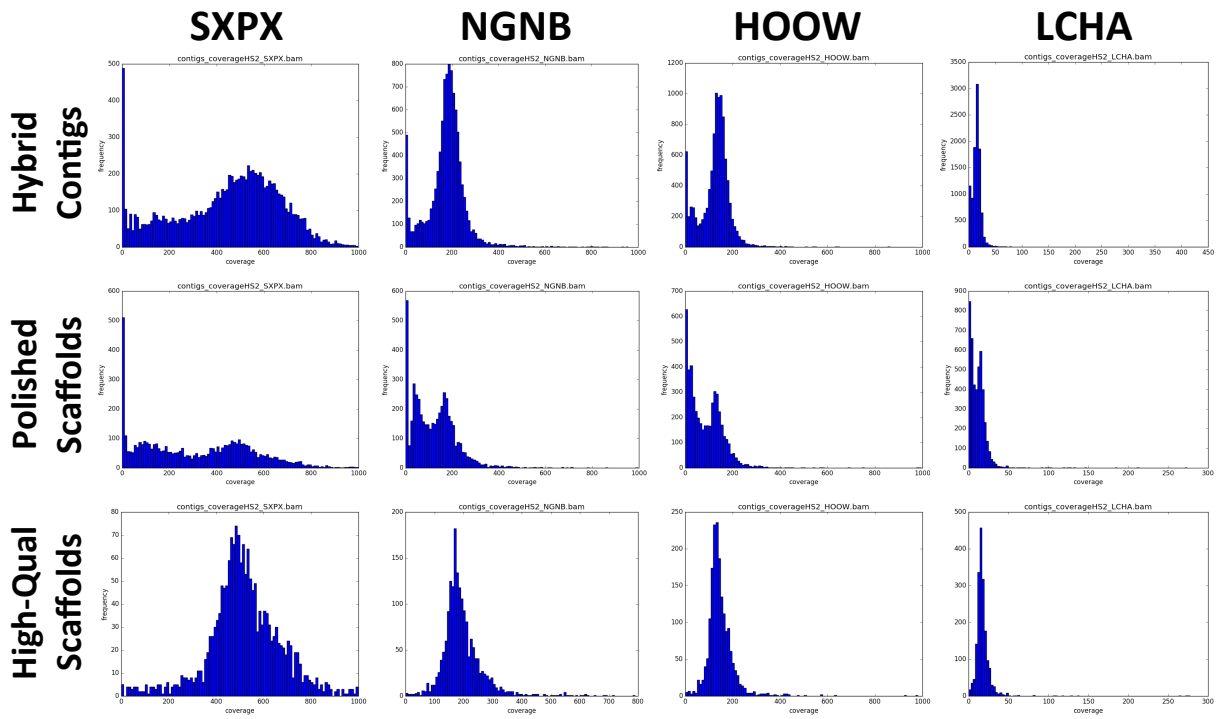


Figure 37. Changes in Illumina sequence coverage throughout assembly pipeline. The Illumina coverage profiles of the assembly throughout the pipeline clearly demonstrate significant changes in the underlying sequence content. It is important to understand how these changes will impact downstream analyses. Improved utilization of coverage profile analyses in the assembly process could help increase assembly quality.

Table 18. Statistics of high-quality sequences after re-scaffolding. After removing the low-quality sequences, a substantial gain was obtained in the scaffolding process. These data suggest that the many small, low-quality fragments contributed to errant alignments that confounded the scaffolding process. After re-scaffolding, the total assembly size was barely increased, but the contiguity was substantially better, as indicated by the N50 statistic and the number of large sequences (>= 1 Mbp).

	High-Quality Scaffolds	Re-scaffolded Scaffolds	0X-Broken Scaffolds
# contigs (>= 1 kbp)	1,972	912	983
# contigs (>= 10 kbp)	1,326	585	656
# contigs (>= 100 kbp)	516	315	366
# contigs (>= 1 Mbp)	4	32	21
Total length (>= 1 kbp)	169,154,875	170,732,603	170,200,492
Total length (>= 10 kbp)	165,918,982	169,201,022	168,668,911
Total length (>= 100 kbp)	137,580,125	160,806,680	159,200,229
Total length (>= 1 Mbp)	4,374,823	42,100,840	26,801,078
Largest contig	1,228,030	2,742,011	2,742,011
GC (%)	50.81%	50.82%	50.82%
N50	260,708	699,849	565,620
L50	191	84	99
# N's per 100 kbp	5,992	7,613	7,324

Table 19. Final statistics of Versions 1.0 and 2.0 assemblies. These data show that the version 2.0 assembly has a slightly higher overall contiguity, as indicated by the N50 statistic. However, it also has fewer small fragments and significantly more large fragments (>= 1 Mbp). The large fragments in particular are important comparative genomics analyses. Ideally, in the future we could further improve the assembly and obtain a small number of chromosome-scale scaffolds.

	Version 1.0 Assembly	Version 2.0 Assembly
# contigs (>= 1 kbp)	2,752	983
# contigs (>= 10 kbp)	998	656
# contigs (>= 100 kbp)	477	366
# contigs (>= 1 Mbp)	4	21
Total length (>= 1 kbp)	184,385,342	170,200,492
Total length (>= 10 kbp)	178,062,322	168,668,911
Total length (>= 100 kbp)	159,204,275	159,200,229
Total length (>= 1 Mbp)	5,632,961	26,801,078
Largest contig	1,870,169	2,742,011
GC (%)	50.83%	50.82%
N50	372,998	565,620
L50	156	99
# N's per 100 kbp	2,501	7,324

2.3.3 Application of Genome Annotation Methods

The utility of genomic sequence is the ability to pinpoint the functional elements that give rise to the properties of the organism for which the genome encodes. There are different types of functional elements in genomic sequences, and many different methods for finding them. The following sections describe the efforts to annotate the *B. braunii* genome in order to obtain useful insights into the contents of the genomic sequences.

2.3.3.1 Prediction of Protein-Coding Genes

One of the major classes of functional genomic elements is the protein-coding gene. Genome annotation largely begins with the identification of protein-coding genes, followed by assignment of biological function (*269*). With the proteome in hand, researchers can do many analyses, such as compare protein families across species, reconstruct metabolic and regulatory networks, etc. Prediction of protein-coding genes is a complex process that involves compiling multiple lines of evidence (*269*). The JGI has well-established genome annotation pipelines in place and was responsible for conducting annotation of the *B. braunii* genome. Appendix A contains a description of the pipeline used by the JGI to predict protein-coding genes in the *B. braunii* genome assemblies.

The "Version 1.0" and "Version 2.0" assemblies of the *B. braunii* genome were both annotated by the JGI pipeline, which enabled a thorough comparison of the assemblies beyond their simple contiguity statistics. The predicted gene set for the "Version 1.0" genome assembly (v1.2) contained 20,577 loci, with 4,274 alternatively spliced transcripts (Table 22). The predicted gene set for the "Version 2.0" genome assembly (v2.1) contained 20,765 loci, with 3,403 alternatively spliced transcripts (Table 22). There were 3,922 loci from the v1.2 annotations that

could not be found in the v2.1 annotations (Table 22). Likewise, there were 3,670 loci from the v2.1 annotations that could not be found in the v1.2 annotations (Table 22). The program BUSCO was used to estimate genome completeness, based on the presence or absence of highly conserved genes (*218*). The v1.2 and v2.1 annotations contained 89.4% and 80.6% of the expected BUSCOs, respectively (Table 22). The size distributions of predicted gene elements (i.e. 5'-UTRs, exons, introns, 3'-UTRs) were compared between the two annotation versions and no substantial differences were found (Figure 21). Both assemblies showed similar distributions of genes across the set of scaffolds (Figure 22). The gene models of each annotation set showed similar degrees of support by EST alignment and peptide homology (Table 23). A comparison of orthologs between *B. braunii* and *C. reinhardtii* with inParanoid (*270*) revealed a slightly larger number of orthlogs in the v1.2 annotation set (Table 22). While the v1.2 annotation set showed a better recovery of BUSCOs, there do not appear to be any other substantial structural or evidential differences between the v1.2 and v2.1 annotation sets.

The recovery of BUSCOs and the higher number of orthlogs with *C. reinhardtii* indicates that v1.2 has a better recovery of gene content, which could be due to the lower gap content of the "Version 1.0" assembly. However, the "Version 2.0" assembly has better assembly contiguity statistics, which is valuable for studies of genome organization and rearrangement. It is possible that further closing of gaps in the "Version 2.0" assembly could increase the completeness of gene content recovery in the annotations. However, it may be better to build new versions of the genome assembly, as algorithms for assembly are constantly evolving and improving. Additionally, the software for annotation is also changing rapidly. The annotations will benefit as software, databases, and ontologies continue to grow. Finally, it is important to note that the genome assemblies are in fact metagenomic, as the DNA did not come from axenic *B. braunii* cultures.

Table 20. Summary of predicted genes for *B. braunii*. These data show that the annotation sets are highly similar with the exceptions of BUSCO recovery and alternative transcripts. This adds important evidence of assembly quality in parallel with the contiguity statistics.

Summary statistics:	v1.2	v2.1
% RNA-seq aligned to genome	81.9%	81.7%
Total transcripts	24,851	24,168
Primary transcripts (loci)	20,577	20,765
Alternative transcripts	4,274	3,403
BUSCO complete %	89.4%	80.6%
n homologs to chlamy by inParanoid	5,656	5,206
n loci failed liftover to other assembly	3,922	3,670

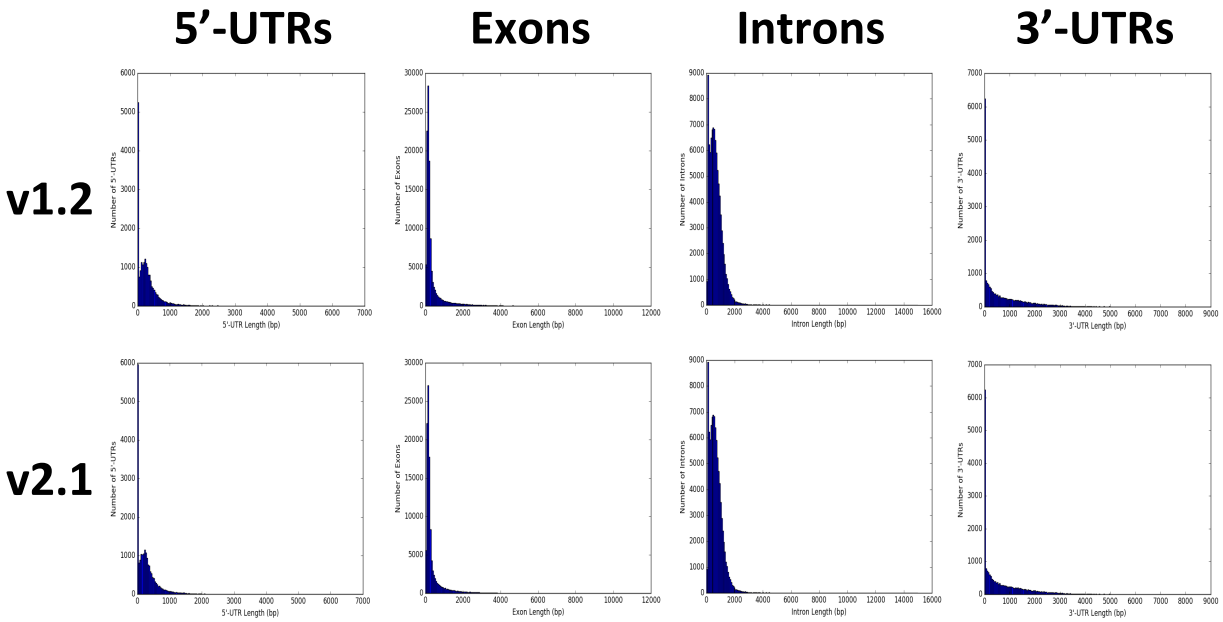


Figure 38. Size distributions of predicted gene elements for *B. braunii*. The distributions of gene element size show minimal differences between the annotation sets. Thus, both assemblies captured similar gene structures.

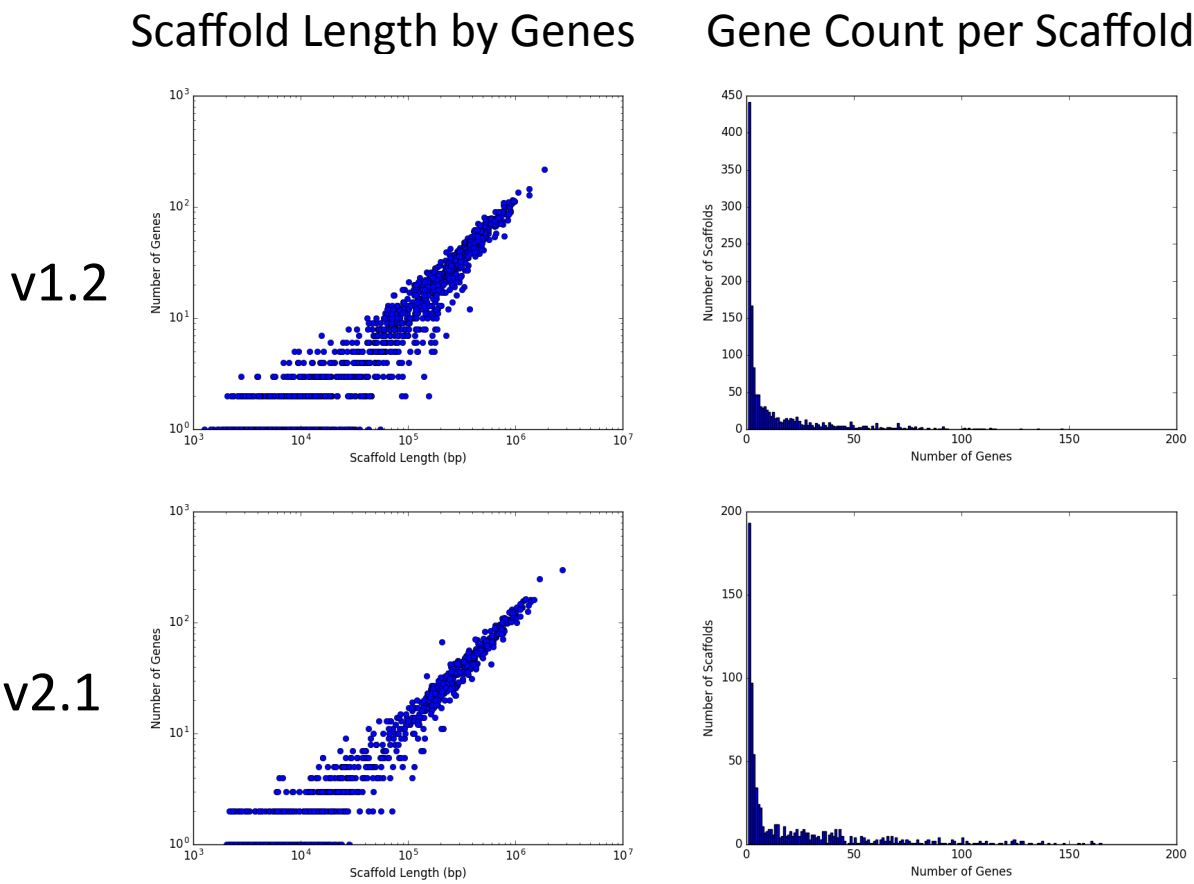


Figure 39. Summary of gene counts per scaffold and by scaffold length. These data show that in both assemblies there are similar relationships between scaffold length and gene count per scaffold. Also, most of the genes are concentrated in groups of 10 or more, with hundreds of scaffolds having little or no gene content.

Table 21. Summary of alignment and homology support per gene model. These data show similar EST support and peptide homology evidence for each of the annotation sets. Both annotation sets have fairly strong evidence supporting the predicted genes.

Gene model support:	v1.2	v2.1
EST support over 100% of their lengths	17,347	17,411
EST support over 75% of their lengths	17,916	18,025
EST support over 50% of their lengths	18,348	18,499
Peptide homology coverage of 100%	163	182
Peptide homology coverage of over 75%	8,518	7,704
Peptide homology coverage of over 50%	12,687	12,217

2.3.3.2 Functional Assignment to Proteins

Comparison of the v1.2 and v2.1 functional annotations reveals that they are highly similar, but each with a bit of unique content (Figure 23). For each annotation, the number of genes with that annotation is fairly similar between v1.2 and v2.1 annotations (Table 24). However, v2.1B is very different, with much fewer genes annotated with EC numbers and KEGG identifiers. Significantly, the KEGG identifiers reported in v2.1B point to pathway objects, while in v1.2 and v2.1 the KEGG identifiers point to orthology objects. The number of genes with GO terms found in v2.1B is higher than in v1.2 and v2.1, but the number of genes with Pfam domains is lower than in v1.2 and v2.1. Additionally, there are more databases found in v2.1B than in either v1.2 or v2.1, including Gene3D, SUPERFAMILY, InterPro, MetaCyc, Reactome, and others. Looking at the number of distinct functions from each database across the annotations, v1.2 and v2.1 are again reasonably similar (Table 25). The v2.1B functions are similar in number to v1.2 and v2.1 in terms of GO and Pfam annotations. The number of EC terms and KEGG terms were not comparable. However, it is possible to map GO and Pfam annotations to KEGG and EC annotations, and thus they could be updated accordingly. This would only require a script to process the information. Nonetheless, these data demonstrate that the method for functional annotation has a substantial impact on the results. Thus, more work is needed to further understand the nature of this impact and improve the accuracy of annotations.

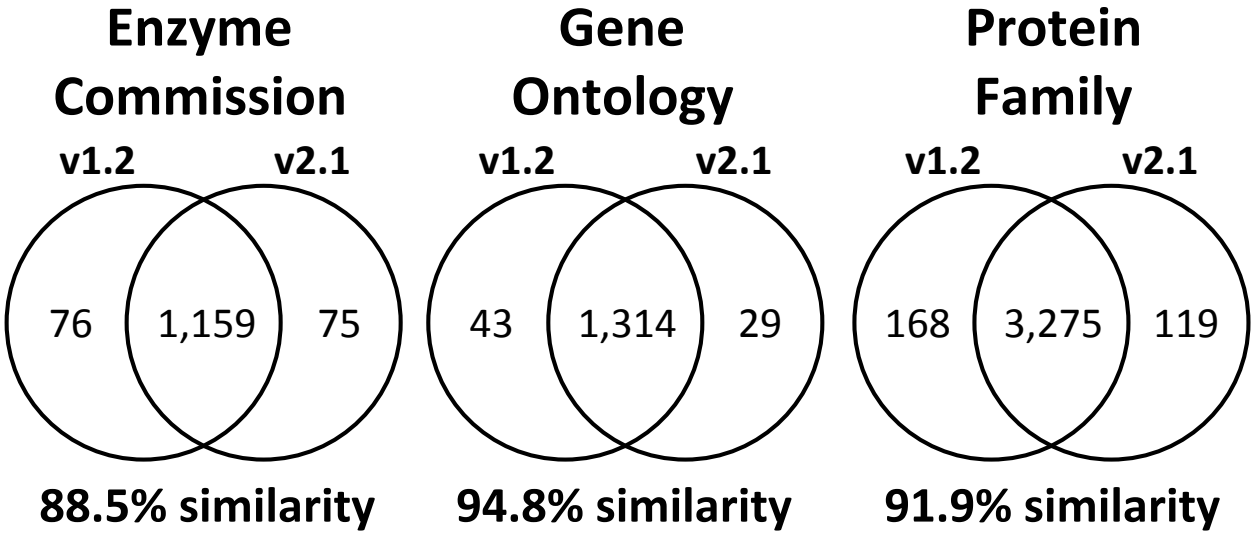


Figure 40. Comparison of functional assignments for *B. braunii* v1.2 and v2.1 proteins. Analysis of the similarity between the two annotation sets using the Jaccard index shows a high degree of similarity. A strong majority of the predicted gene functions are agreed upon by both annotation sets.

Table 22. Number of genes annotated with each database. There was some difficulty in reproducing the results of the gene annotations from JGI. A similar number of GO and Pfam predictions was obtained, but there was great variance in the EC and KEGG predictions.

Annotated genes:	v1.2	v2.1	v2.1B
n gene w EC	4,267	4,211	885
n gene w GO	5,936	5,829	6,507
n gene w KEGG	3,604	3,606	885
n gene w KOG	4,288	3,713	NA
n gene w Pfam	8,927	8,696	8,237

Table 23. Number of distinct functions from each database. These data show that the GO and Pfam predictions were reproducible, but the EC and KEGG predictions were not. It could be possible to convert the Pfam and GO annotations into synonymous EC and KEGG annotations.

Distinct annotations:	v1.2	v2.1	v2.1B
n distinct EC	1,236	1,234	428
n distinct GO	1,357	1,343	1,520
n distinct KEGG	2,853	2,845	118
n distinct KOG	2,394	2,153	NA
n distinct Pfam	3,425	3,376	3,339

2.3.3.3 Prediction of Repetitive Elements

Genomic repeat content is incredibly difficult to assemble and contributes to fragmented genome assemblies. Different repeat assembly heuristics of various assembly algorithms can lead to substantially different assembly results. The repeat content was assessed in the *B. braunii* "Version 1.0" and "Version 2.0" assemblies using RepeatMasker (*271*). The "Version 1.0" assembly has 37.41% repeat content, while the "Version 2.0" assembly has 29.69% repeat content (Table 26). This is the most significant difference between the two assemblies. Since the total size of the "Version 2.0" assembly (170.2 Mbp) is closer to the estimated genome size (166.2 Mbp), it stands to reason that the true repeat content is close to 30%, and that the "Version 1.0" assembly (184.3 Mbp) has an overrepresentation of repeats in the assembly. The main difference between the two versions is the amount of interspersed repeats (i.e. transposable elements). The amount of simple sequence repeats is not very different between the two versions.

Table 24. Summary of predicted repeat elements. These data show the major difference in repeat contents between "Version 1.0" and "Version 2.0" of the *B. braunii* genome. In particular, the interspersed class of repeats is enlarged in "Version 1.0". Simple repeat sequences constitute less than 10% of the genome.

	Genome V1	**Genome V2**
Genome size (Mb)	184.3	170.2
Bases masked	37.41%	29.69%
Repeat families	542	496
Interspersed Repeats		
SINEs	0.00%	0.05%
LINEs	1.56%	1.42%
LTR elements	13.89%	8.04%
DNA elements	0.99%	2.06%
Unclassified	11.18%	8.94%
Other Repeats		
Small RNA	0.07%	0.02%
Satellites	0.00%	0.00%
Simple repeats	8.92%	8.41%
Low complexity	0.86%	0.80%

2.3.3.4 Prediction of DNA Methylation

One of the great utilities of PacBio sequencing data is that in addition to use for assembly, gap closing, and polishing, it can be used to detect DNA base modifications, such as methylation (*272*). The modification of DNA bases is incredibly complex, is not limited to methylation, and is emerging as an entirely new language that we are just beginning to understand (*273*). In microalgae, it is well established that DNA methylation exists and plays a role in gene silencing (*274*). In order to test for DNA base modification in *B. braunii*, the PacBio data was aligned against the "Version 2.0" assembly with BLASR. The alignments were then processed with kineticsTools (available at https://github.com/PacificBiosciences/kineticsTools, unpublished), a program developed by PacBio to detect DNA modifications. There were 3,367,675 modified bases (2% of the total bases) detected in the "Version 2.0" assembly. The number of modified bases per scaffold was calculated and then plotted against the scaffold length (Figure 24). There is a strong linear correlation between scaffold length and the number of modified bases. This indicates that the methylation marks are well distributed throughout the genome. Although this data is valuable, it would be much better if methylation were analyzed under dynamic conditions. Such data would enable the analysis of DNA modifications over time and in response to stimuli. It may also be interesting to explore correlations between DNA modifications and gene expression patterns.

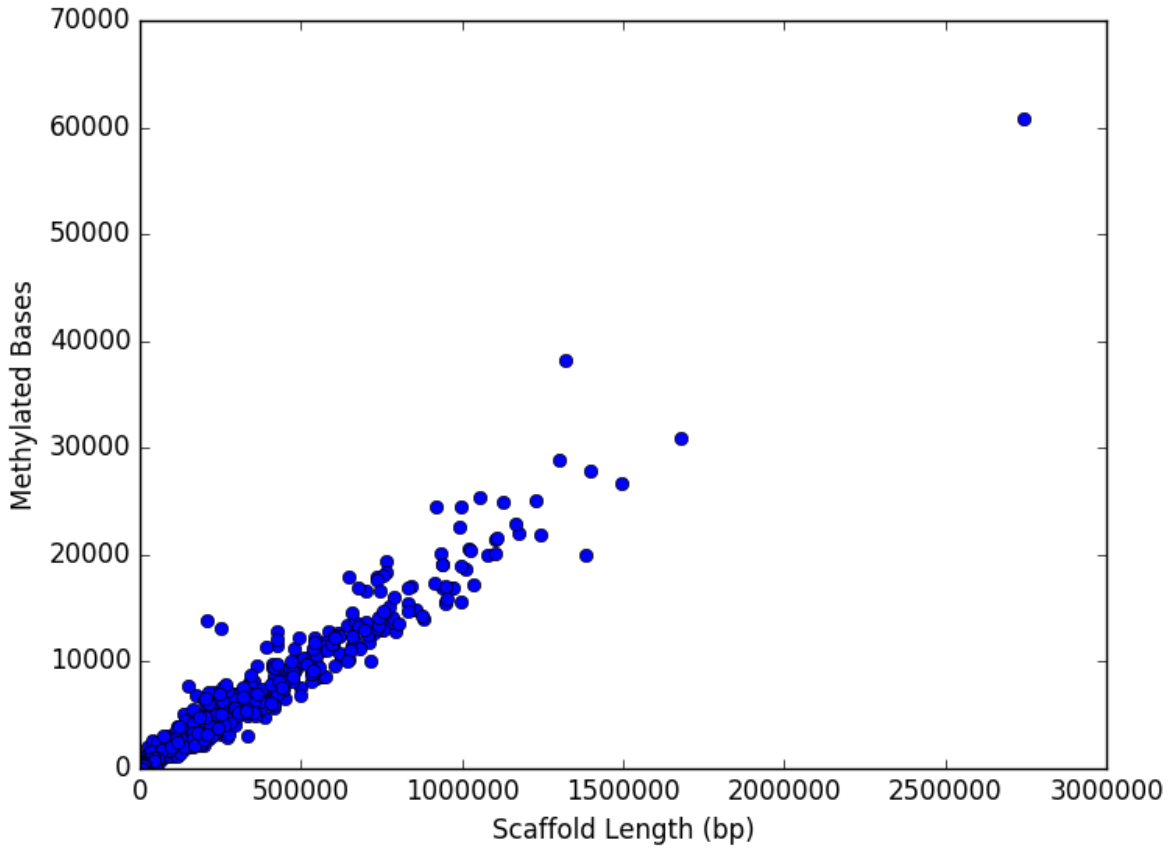


Figure 41. Correlation between scaffold length and number of methylation marks. These data show that scaffold length correlates strongly with the number of methylated bases in the scaffold. This indicates that methylation is well distributed throughout the genome, with some variance. The experiment proves that DNA methylation can be detected in *B. braunii* with PacBio sequencing.

2.4 Conclusion

This work has demonstrated various experimental approaches to genome assembly, highlighting some of their limitations. In an effort to improve the state of the art, new methods of genome assembly were developed and used to reconstruct the *B. braunii* genome. While some improvements were made, there are still substantial barriers to achieving a reference-quality assembly with chromosome-scale scaffolds. New sequencing platforms and assembly algorithms will soon enable big steps towards assembling a reference-quality genome for *B. braunii*. Future efforts should focus on continuing to refine the genome assembly and obtaining new types of sequencing data.

Although the current version of the *B. braunii* genome is still quite fragmented, it nonetheless captures a large amount of the gene content. These predicted genes can be functionally annotated and are enormously informative. They enable the analysis of different pathways and the comparison with other species. Repeat elements and DNA methylation were also annotated within the genome, adding more layers of information, and opening up new possibilities. The databases and algorithms used in genome annotation are rapidly evolving. As more information accumulates and gene models improve, functional annotations will become better. Future efforts should include periodically updating the genome annotations.

Both the genome assembly and the functional annotations pose management challenges because of the maintenance required to keep them updated. Who is responsible for this maintenance? How will it be funded, documented, distributed? But if the maintenance is made, then the field of *B. braunii* research will benefit from the improved information.

3. COMPARATIVE GENOMICS OF VIRIDIPLANTAE

The work described in this section takes a two-pronged approach to elucidating evolutionary, structural, and functional relationships between the sequenced and annotated species of Viridiplantae that are contained in the publicly available Phytozome database. The goal of this work is to demonstrate new types of systems-level comparative analyses computed from the genome annotations.

3.1 Introduction

Following is first an overview of the Phytozome database and some data describing the genomes of the species within the database. This provides important context for results and discussion presented later in the section. Second is an overview of the most relevant gene and protein annotation systems. This information is essential for understanding the work presented in this section.

3.1.1 Survey of Assembled Viridiplantae Genomes

In 2012, the JGI simultaneously released the Genome Portal (*275*) and the Phytozome database (*276*). The Genome Portal was built to facilitate user access to the massive amounts of DNA sequencing and analysis data that were being generated annually at the JGI. The goal was to provide portals for each organism, with analysis tools, a genome browser, annotation tools, protein pages, and links to other JGI resources. At the time of this writing, there are 76,317 publicly available project entries in the JGI Genome Portal. Managing such a large amount of data is a very substantial challenge, and currently there is some data redundancy and clutter. Improvements in data management and sharing will result in greater accessibility and facilitate further analyses.

Phytozome, focused on green algae and land plants, was built to store the increasing number of sequenced Viridiplantae genomes, provide access to the sequences and functional annotations, and give researchers a set of tools for comparative analyses. Phytozome version 12 has genome sequences and annotations for 64 "standard release" species and a number of additional "early release" species (Figure 25). There is a large range in genome size among the Viridiplantae, from *Ostreococcus tauri* at 12.6 Mbp (*277*) to *Zea mays* at 2.3 Gbp (*278*), with the median genome size about 400 Mbp (Figure 26). The GC contents of each species was assessed with QUAST (*279*), revealing that it ranges from 10-80% across the sample of genomic windows (Figure 27). One of the most important aspects of the Phytozome database is that the genomes within it were subjected to standardized gene prediction and functional annotation pipelines. In order to compare predicted genes and functions across multiple species, it is essential that the methods for annotation are consistent.

3.1.2 Review of Functional Annotation Systems

Functional annotation of predicted genes is dependent upon structured languages describing their ontological features. There are different languages currently available to describe various biochemical and biological roles for genes and proteins. One of the earliest biochemical languages developed was the enzyme nomenclature system, which has roots in the 19th century (*280*). It was in the 1950s that the current enzyme classification system began to coalesce. In 1962, the first report from the Enzyme Commission (EC) appointed by the International Union of Biochemistry (IUB) was comprehensively summarized by the then Secretary-General of the IUB (*281*). The EC recommended classifying enzyme activity into six main groups: 1) oxidoreductases, 2) transferases, 3) hydrolases, 4) lyases, 5) isomerases, and 6) ligases. Within each group are three

layers of sub-groups defining increasing levels of reaction specificity. For example, the EC number 2.7.2.2 describes the enzyme carbamate kinase, which catalyzes the synthesis of carbamoyl phosphate from ATP, bicarbonate, and ammonia. The EC system of nomenclature continues to undergo refinement, but has not been fundamentally altered for nearly 30 years (*282*). During this time, numerous other languages have developed. The Kyoto Encyclopedia of Genes and Genomes (KEGG) is a comprehensive database with a structured language describing molecular functions of genes and proteins, ortholog groups, functional hierarchies, metabolic pathways, and more (*283*). The Gene Ontology (GO) is a tripartite structured language describing molecular functions, biological processes, and cellular components, each with sets of defined terms that are related through directed acyclic graphs (*284*). Statistical tests have been developed for determination of significant differences between two or more gene sets annotated with GO terms (*285*). The Pfam database contains groups (i.e. families) of related proteins derived from multiple sequence alignments using hidden Markov models (HMMs), revealing common domain structures and sequence elements, and currently has 16,712 families and 604 clans (*286*). Each one of these databases and languages has unique elements, but they also have overlapping information and certain terms can be translated across ontologies. Fortunately, there are EC, KEGG, GO, and Pfam annotations available for all of the genomes in Phytozome. This data makes it possible to conduct large-scale whole genome comparisons based on functions.

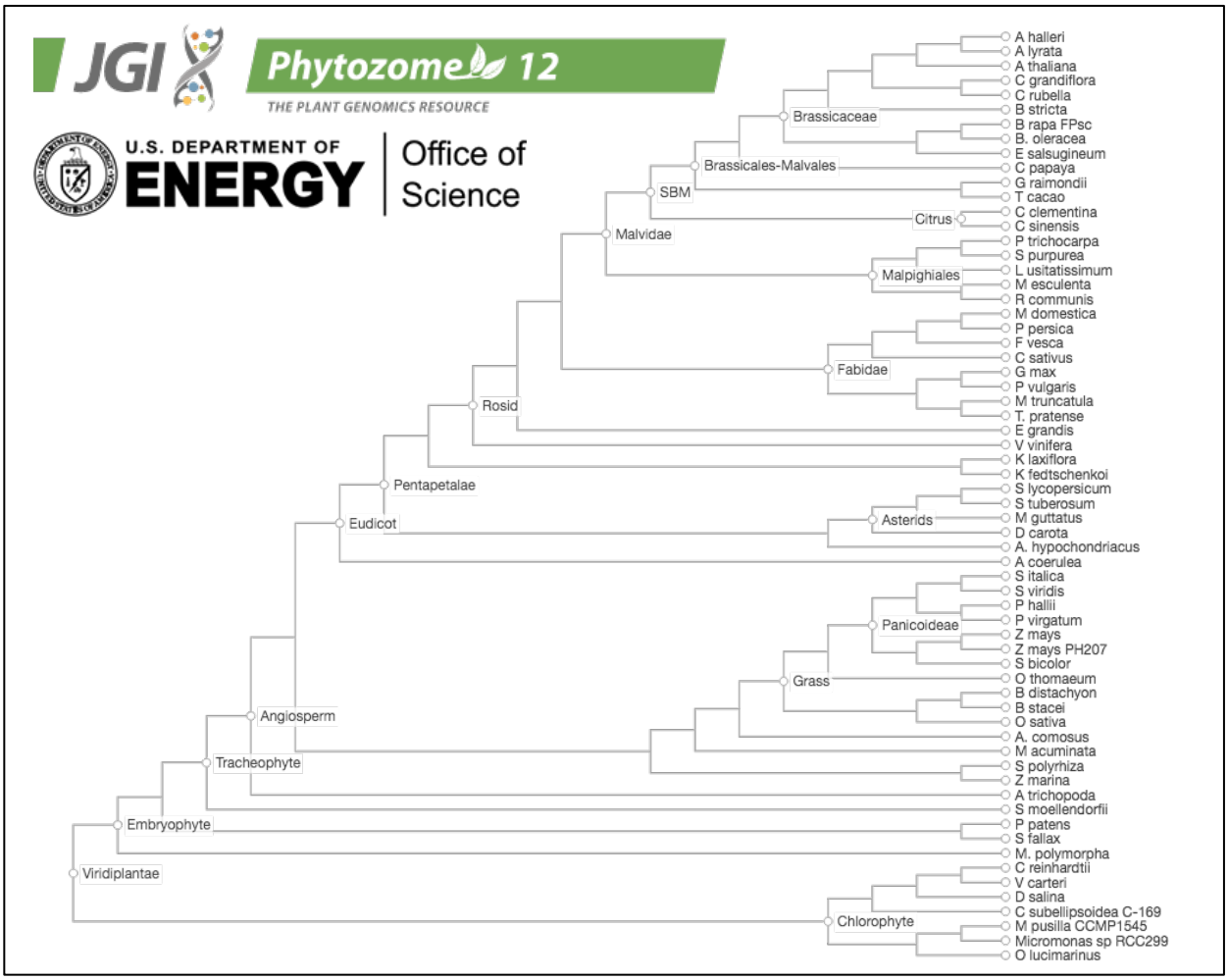


Figure 42. Phylogenetic tree of Viridiplantae genomes in Phytozome. This figure from the Phytozome website shows the overall phylogenetic classifications of the many species contained in the database. However, this tree does not include the latest additions to the database, which now exceeds 90 species.

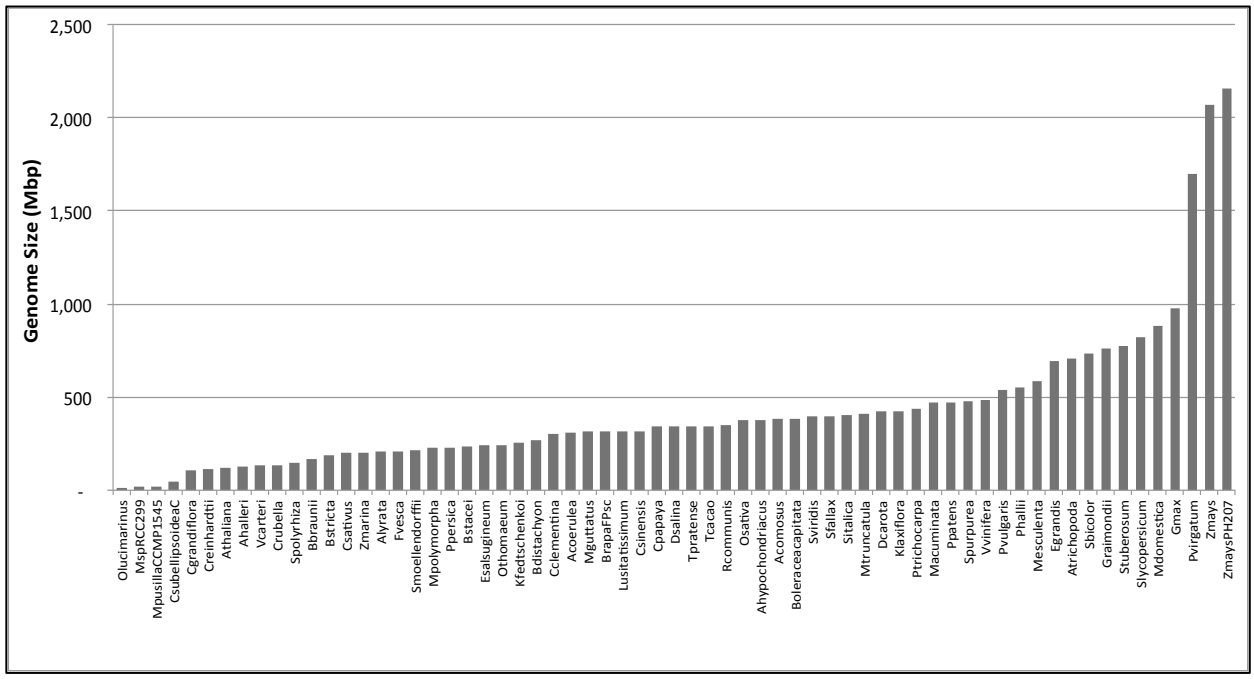


Figure 43. Variation in sizes of Viridiplantae genomes. These data show the range of genome sizes found in the Phytozome database, ranging from roughly 10 Mbp to 2 Gbp. However, the median genome size is approximately 400 Mbp, with only a handful of species greatly exceeding the median. The species are sorted from smallest genome to largest.

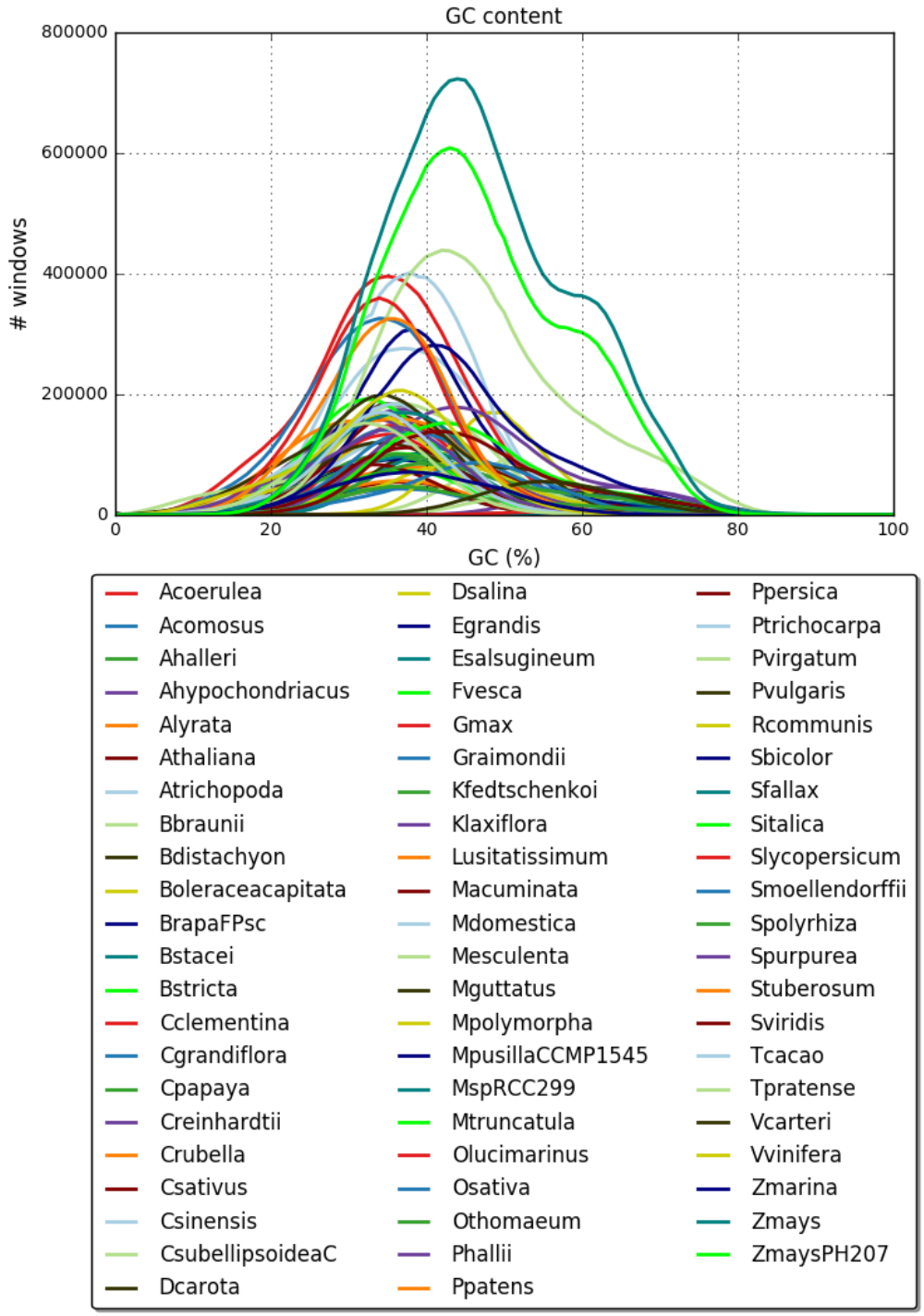


Figure 44. QUAST analysis of GC content in Viridiplantae genomes. These data show the wide range of GC contents found in each species within the Viridiplantae. Some species have a narrow range of GC contents, while other species have a very wide range. There is no apparent correlation between genome size and GC contents. Most species show a peak in GC contents around 30-40%.

3.2 Materials and Methods

All analyses conducted in the course of this work were based on information contained in the publicly available database Phytozome v12 (https://phytozome.jgi.doe.gov/pz/portal.html). Computational methods associated with the processing and analysis of the database are described in detail in Appendix B.

3.3 Results and Discussion

This work begins by analyzing the genome annotations of all the species available in the Phytozome database on a global basis. The high-level functional and structural comparisons serve to illuminate broad evolutionary trends and relationships. This work is followed by specific and detailed analyses of several key biological processes using the KEGG ontology.

3.3.1 Functional Signatures in Genome Annotations

In order to obtain a global picture of the distribution of functional annotation terms across Viridiplantae, for each species the number of times was counted that a given term occurred in the annotations. The counts for each species were joined together into a table containing the counts for all terms in all species. This "term counting" approach was applied separately to the EC, GO, KEGG, and Pfam annotations. The resulting tables were log-transformed and then visualized and hierarchically clustered with Morpheus (https://software.broadinstitute.org/morpheus/) (Figures 28-31). These clustered heatmaps yield visual insight into the complete functional signature for each species. They enable quick identification of features that differentiate individual species or groups of species (i.e. clades). For example, there are clearly functional regions that are low copy-number or missing in Chlorophyta, but enriched in Embryophyta, and vice versa. The dendrogram

of columns obtained by hierarchical clustering strongly approximates the known phylogenetic tree. Interestingly, the different ontologies give slightly different column dendrograms. The ontologies also show different patterns of term distribution in the annotation sets. For example, the KEGG dataset has low term redundancy and many unique or low frequency terms (Figure 28), whereas the GO dataset has a subset of terms with very high frequencies in the annotations (Figure 31).

One of the simplest operations that can be done with this type of data is to select values from the table based on specific criteria, such as minimum term frequency, occurrence or absence in certain species, etc. Several such operations were applied to the data. Terms were selected that were found only in *B. braunii* among the Viridplantae, found only in *B. braunii* among the Chlorophyta, were missing only in *B. braunii* among the Viridiplantae, were missing only in *B. braunii* among the Chlorophyta, were found in all Chlorophyta species, were found in all Embryophyta species, were found in all Viridiplantae species, and were found in any species of Viriplantae (Table 27). Furthermore, the set of functions found only in *B. braunii* among the Chlorophyta was subtracted from the set of functions found only in *B. braunii* among the Viridiplantae, giving a set of functions that *B. braunii* shares with Embryophyta and not Chlorophyta (Table 27). The KEGG terms that were missing only in *B. braunii* among the Viridiplantae were mapped to the BRITE hierarchy to obtain higher-level classifications of the terms (Table 28). The most abundant categories with missing functions were exosome, ribosome biogenesis, membrane tracking, chromosome, and proteasome. Similarly, the KEGG terms found only in *B. braunii* among the Viridiplantae were mapped to BRITE, revealing innovations in the ubiquitin system, cytochrome P450s, peptidases, cytoskeleton proteins, and others (Table 29). Finally, the KEGG terms that *B. braunii* shares with Embryophyta and not Chlorophyta were mapped to BRITE, revealing functions in the categories of photosynthesis proteins, chromosome,

ubiquitin system, mitochondrial biogenesis, and others (Table 30). These data demonstrate the power of taking a "top-down", global approach to analyzing functional annotations for a group of species. Building on this work, the application of advanced statistical methods should help to improve the insights obtained through analysis of global functional signatures. The KEGG hierarchy is also particularly useful for augmenting the analytical pipeline to extract specific pieces of information regarding pathways of interest. The filtering and selection that can be applied with the KEGG framework will enable standardized comparisons of pathway structure across the species. However, there is the limitation that KEGG may not have adequate terms to describe the pathways in all of the species. Divergent pathway evolution and structure could complicate this analysis, due to the potential lack of terms to describe uniquely evolved functions in certain species. Using KEGG to compare pathways across species without a complete inventory of all possible parts of the pathway would lead to hidden components missing from the results. Thus, while standardization of terms in KEGG is important, it is equally important to have detailed studies of pathways in separate species and flexible terminology to describe such features.

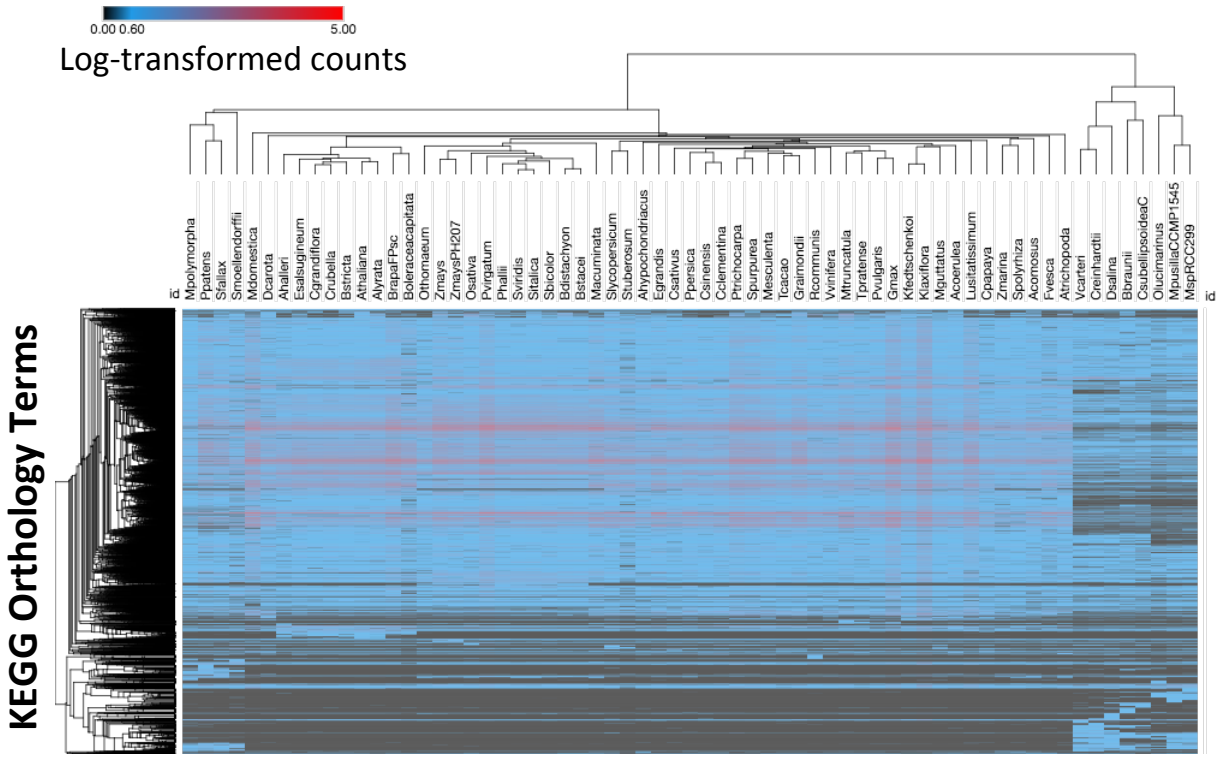


Figure 45. Functional signatures of Viridiplantae with KEGG. These data show that KEGG terms have a high degree of uniqueness, with few repeated terms. The dendrogram of species created by hierarchically clustering the columns shows clear evolutionary relationships.

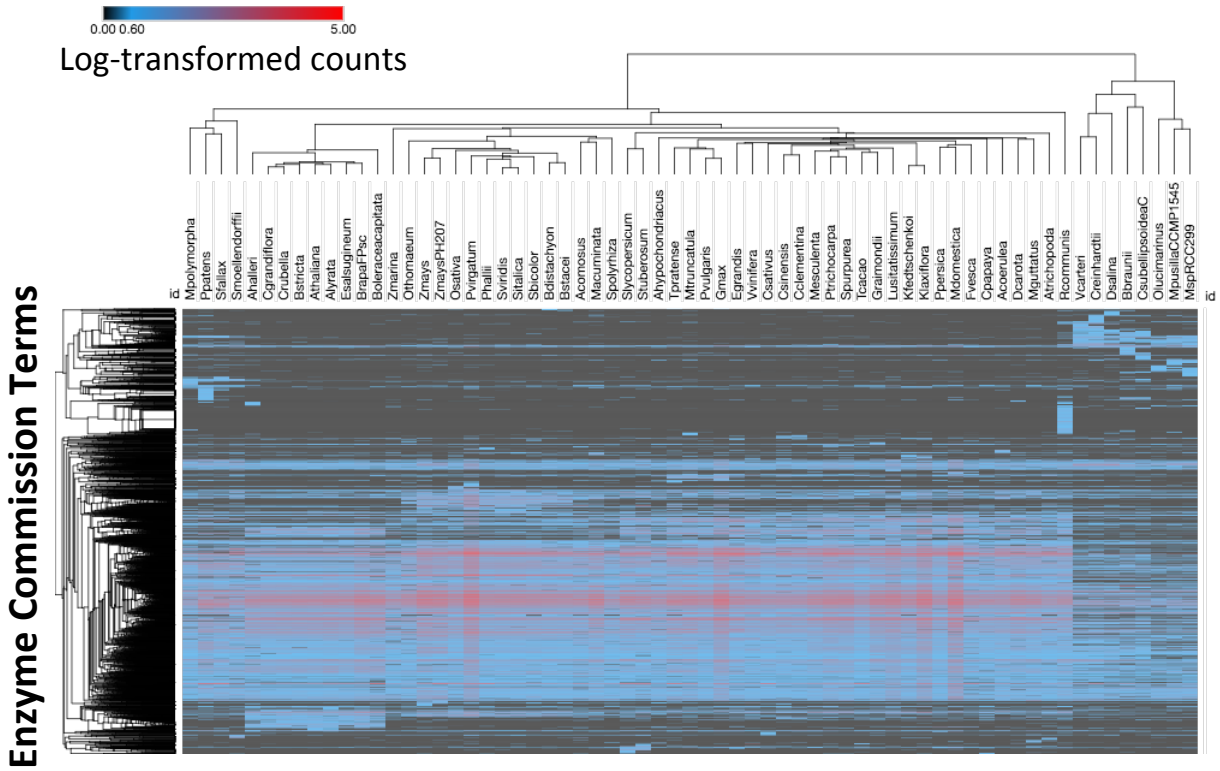


Figure 46. Functional signatures of Viridiplantae with EC. These data show a large degree of variation amongst all the species, with a fairly sparse matrix. However, there are also clearly core biochemical reactions that clearly span the entire set of species. In the Chlorophyta, there is very little redundancy of reactions, but certain reactions in the Embryophyta are performed by multiple genes.

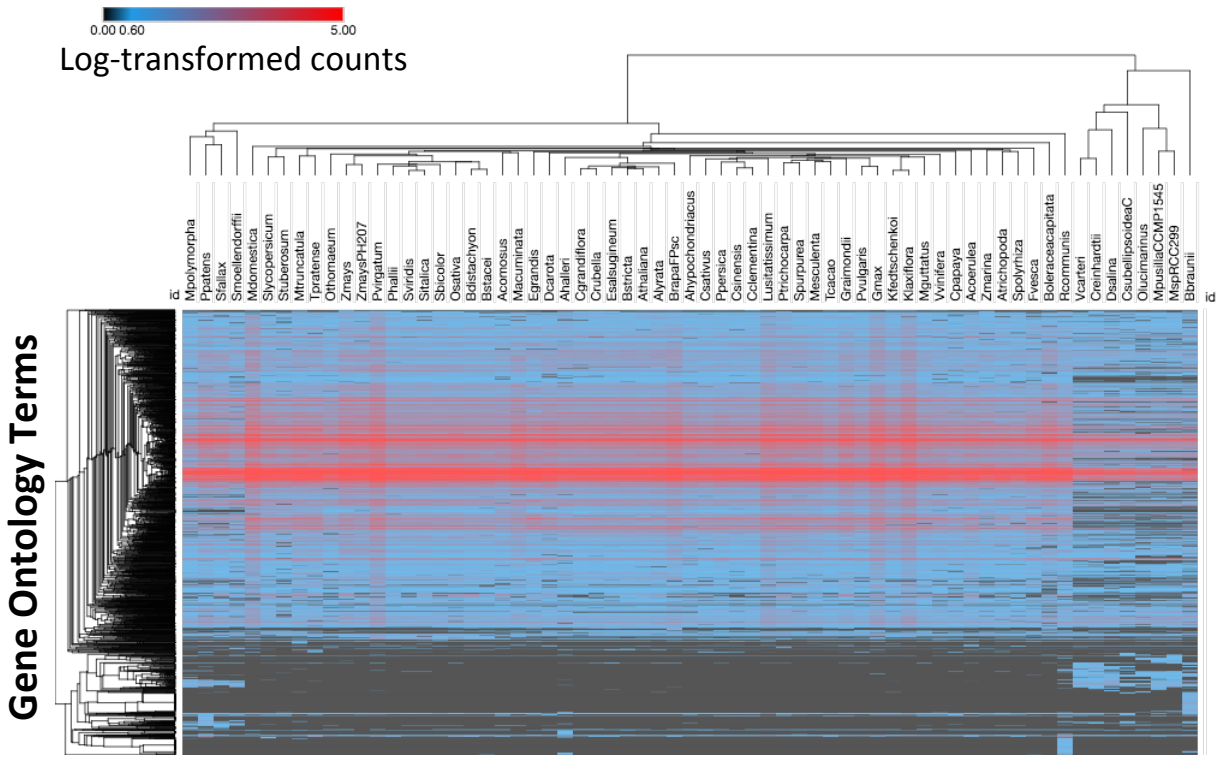


Figure 47. Functional signatures of Viridiplantae with GO. These data show a large sub-set of highly conserved functions, many of which are amplified in frequency within the Embryophyta, as compared to the Chlorophyta. Nonetheless, the matrix is fairly dense, indicating that many of the terms are part of core cellular processes.

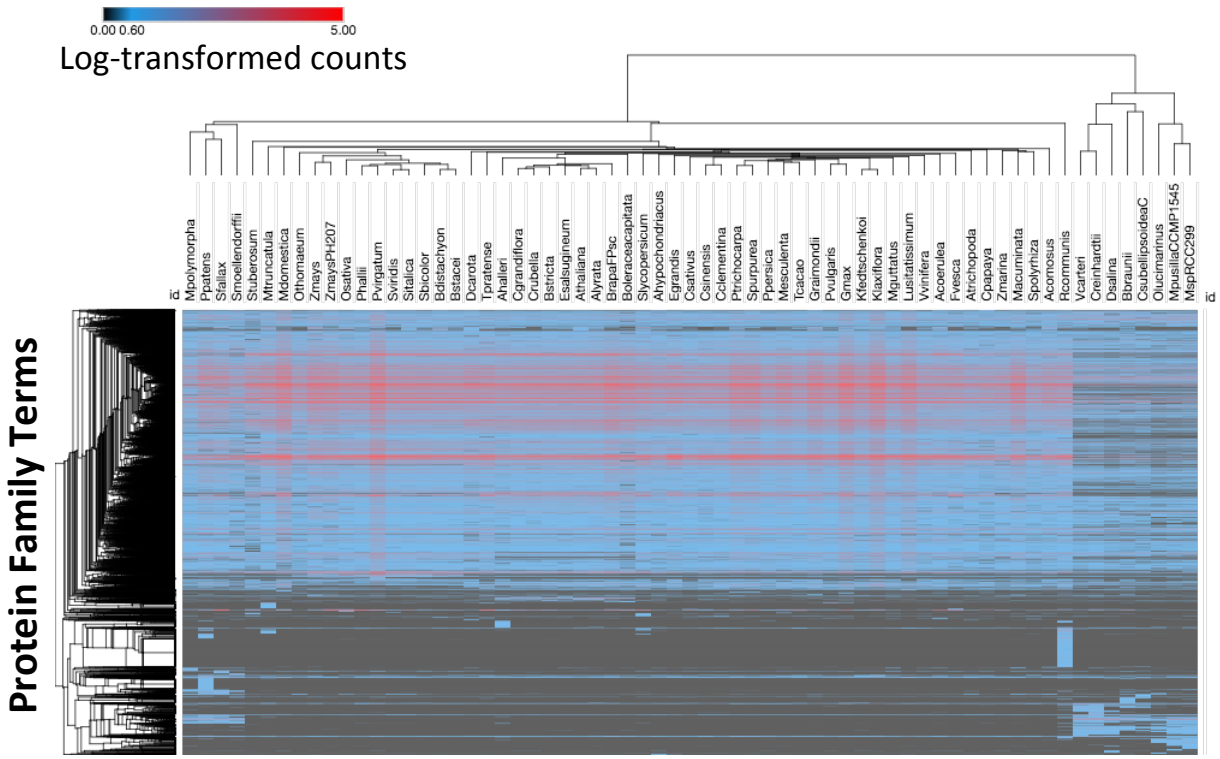


Figure 48. Functional signatures of Viridiplantae with Pfam. These data show a high degree of conserved domains in the Pfam annotations. There are some domains that become highly duplicated in the Embryophyta, but there is minimal duplication in the Chlorophyta. As with the other annotations, there are clear, unique signatures for several groups of species.

Table 25. Summary of functional term selections from each ontology. The power of the annotation database created in this work is the ability to select terms from it according to specific criteria. These data show how useful information can be extracted from the databases to provide insights about evolution of individual species or sub-groups of species.

	EC Terms	GO Terms	KEGG Terms	Pfam Terms
found only in *B. braunii* among Viridiplantae	22	80	37	48
found only in *B. braunii* among Chlorophyta	70	105	98	129
missing only in *B. braunii* among Viridiplantae	28	39	38	80
missing only in B. braunii among Chlorophyta	48	64	76	158
found in all Chlorophyta species	650	877	1,496	1,889
found in all Embryophyta species	815	1,014	1,603	2,229
found in all Viridiplantae species	503	722	905	1,364
found only in *B. braunii* and Embryophyta	48	25	61	81
found in any species of Viridiplantae	2,991	2,049	4,598	6,474

Table 26. BRITE mapping of KEGG terms missing only in *B. braunii.* These data show terms that were found in all species of Viridiplantae except for *B. braunii*. This could indicate significant evolutionary gene losses. However, they could also be absent from the genome due to incompleteness (i.e. gaps) in the genome assembly.

BRITE	Description	Terms
ko04147	Exosome	6
ko03009	Ribosome biogenesis	4
ko04131	Membrane trafficking	4
ko03036	Chromosome	4
ko03051	Proteasome	4
ko03019	Messenger RNA biogenesis	3
ko03041	Spliceosome	3
ko03110	Chaperones and folding catalysts	2
ko01006	Prenyltransferases	2
ko02000	Transporters	2
ko03016	Transfer RNA biogenesis	2
ko04031	GTP-binding proteins	2
ko03021	Transcription machinery	2
ko03029	Mitochondrial biogenesis	1
ko03011	Ribosome	1
ko01004	Lipid biosynthesis proteins	1
ko00536	Glycosaminoglycan binding proteins	1
ko01009	Protein phosphatases and associated proteins	1
ko01002	Peptidases	1
ko04121	Ubiquitin system	1
ko04090	CD molecules	1

Table 27. BRITE mapping of KEGG terms found only in *B. braunii*. These terms are those found in no species of Viridiplantae except *B. braunii*. They indicate important systems that could contribute to the unique morphology and physiology of the species. However, it is possible that there are contaminating metagenomic sequences that influence these results.

BRITE	Description	Terms
ko04121	Ubiquitin system	5
ko00199	Cytochrome P450	4
ko01002	Peptidases	4
ko00002	KEGG modules	4
ko04812	Cytoskeleton proteins	3
ko04516	Cell adhesion molecules and their ligands	2
ko00536	Glycosaminoglycan binding proteins	2
ko03036	Chromosome	2
ko00535	Proteoglycans	2
ko01001	Protein kinases	1
ko04091	Lectins	1
ko02000	Transporters	1
ko04131	Membrane trafficking	1
ko03019	Messenger RNA biogenesis	1
ko03029	Mitochondrial biogenesis	1
ko03400	DNA repair and recombination proteins	1
ko01009	Protein phosphatases and associated proteins	1
ko03009	Ribosome biogenesis	1

Table 28. BRITE mapping of *B. braunii* KEGG terms shared with Embryophyta and not Chlorophyta. These data show pathways where *B. braunii* shares annotation terms with the Embryophyta, but not the other Chlorophyta. These are potential examples of convergent evolution, where similar functions have unfolded in separate lineages.

BRITE	Description	Terms
ko00194	Photosynthesis proteins	12
ko03036	Chromosome	7
ko04121	Ubiquitin system	6
ko03029	Mitochondrial biogenesis	4
ko04147	Exosome	4
ko03011	Ribosome	4
ko01001	Protein kinases	3
ko03400	DNA repair and recombination proteins	3
ko03041	Spliceosome	3
ko03110	Chaperones and folding catalysts	2
ko01003	Glycosyltransferases	2
ko03021	Transcription machinery	2
ko03000	Transcription factors	1
ko03019	Messenger RNA biogenesis	1
ko03032	DNA replication proteins	1

3.3.2 Evolution of Gene Organization in Genomes

Aside from the functional perspective of comparative genomics, there is the physical perspective. That is, the structural arrangement and spatial organization of the sequence elements (i.e. genes, repeats, etc) across chromosomes. This information can be determined by analyzing the sequence coordinates of annotations in the GFF3 files that accompany the genome assembly. This work is dependent upon high-quality assemblies, with scaffolds approximating the chromosomes. If an assembly consists of too many fragments, the higher order information is lost. The degree of contiguity for all of the Viridiplantae assemblies was assessed with QUAST, showing that many of the assemblies are fairly contiguous, but some of them are highly fragmented (Figure 32). GenHub (https://github.com/standage/genhub) is an open-source toolkit to explore genome composition and organization, taking genome sequences, protein sequences, and gene coordinates as input. The output is a set of calculated features, which enable quantitative comparisons of various features across multiple species. To compare genome size with feature size, several boxplots were constructed, with the species sorted along the x-axis from smallest genome to largest (Figures 33-35). Across the Viridiplantae, gene sizes remain relatively constant, compared to the sizes of intergenic regions, which increase consistently with genome size (Figure 33). Interestingly, the Chlorophyta, notably *Dunaliella salina*, have larger genes than the Embryophyta. Looking at genes, exon length remains highly consistent across Viridiplantae, with the exceptions of *Ostreococcus* and *Micromonas*, which almost entirely lack introns (Figure 34). These data suggest that introns are largely responsible for driving expansions in gene length. Looking at transcripts, 5'-UTR and CDS lengths remain consistent across Viridiplantae, while there is greater variation in 3'-UTR lengths (Figure 35). Two exceptions in CDS lengths are *C. reinhardtii* and *V. carteri*, which have much longer CDSs than all of the other Viridiplantae. The

Chlorophyta have longer 5'-UTRs than the Embryophyta, with *V. carteri, B. braunii*, and *C. reinhardtii* having the longest. The case of expanded CDS length in the two Chlorophytes *C. reinhardtii* and *V. carteri* is particularly interesting, given the very high degree of conservation in CDS length outside of these two species. Another topic of interest is that the Chlorophyta have longer 5' and 3' UTRs than the Embryophyta. This could be an indication of evolutionary developments in regulatory mechanisms tied to UTRs. One would expect that the Embryophyta would have more complex UTR regulatory processes, given the number of highly specific functions that land plants have evolved. While these data are informative, they are limited by the lack of chromosome-scale genome assemblies. Even assemblies consisting of 500-1,000 contigs are missing a great deal of higher-level structural information. Gene order in the chromosomes is very important for studying genome rearrangements in populations and across time.

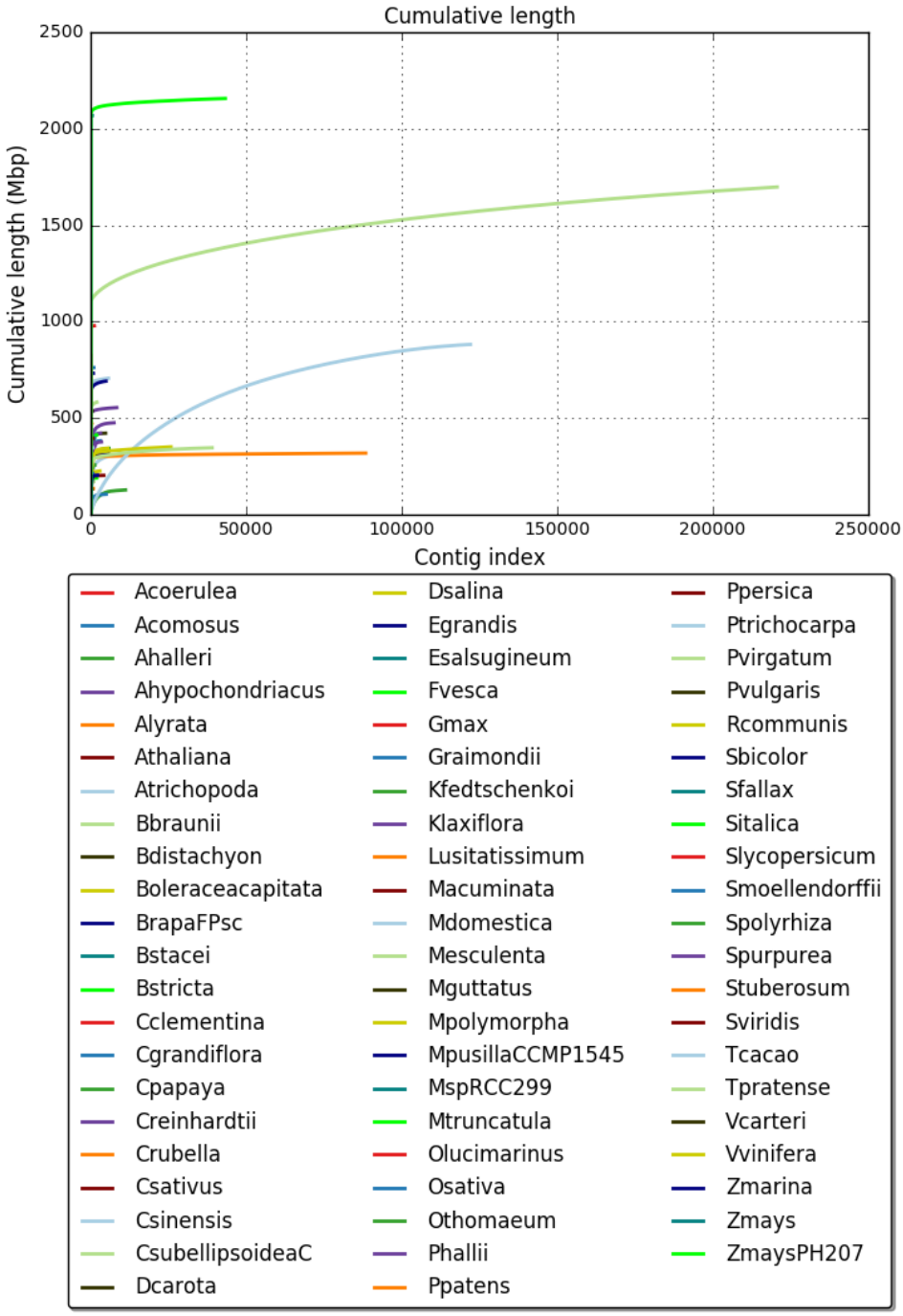


Figure 49. QUAST analysis of Viridiplantae assembly contiguity. There is a fairly large range in genome assembly quality within the Phytozome database. Only a handful of assemblies have a very high degree of contiguity. The majority of assemblies have at least 1,000 sequences. Another handful of assemblies are highly fragmented, low-quality.

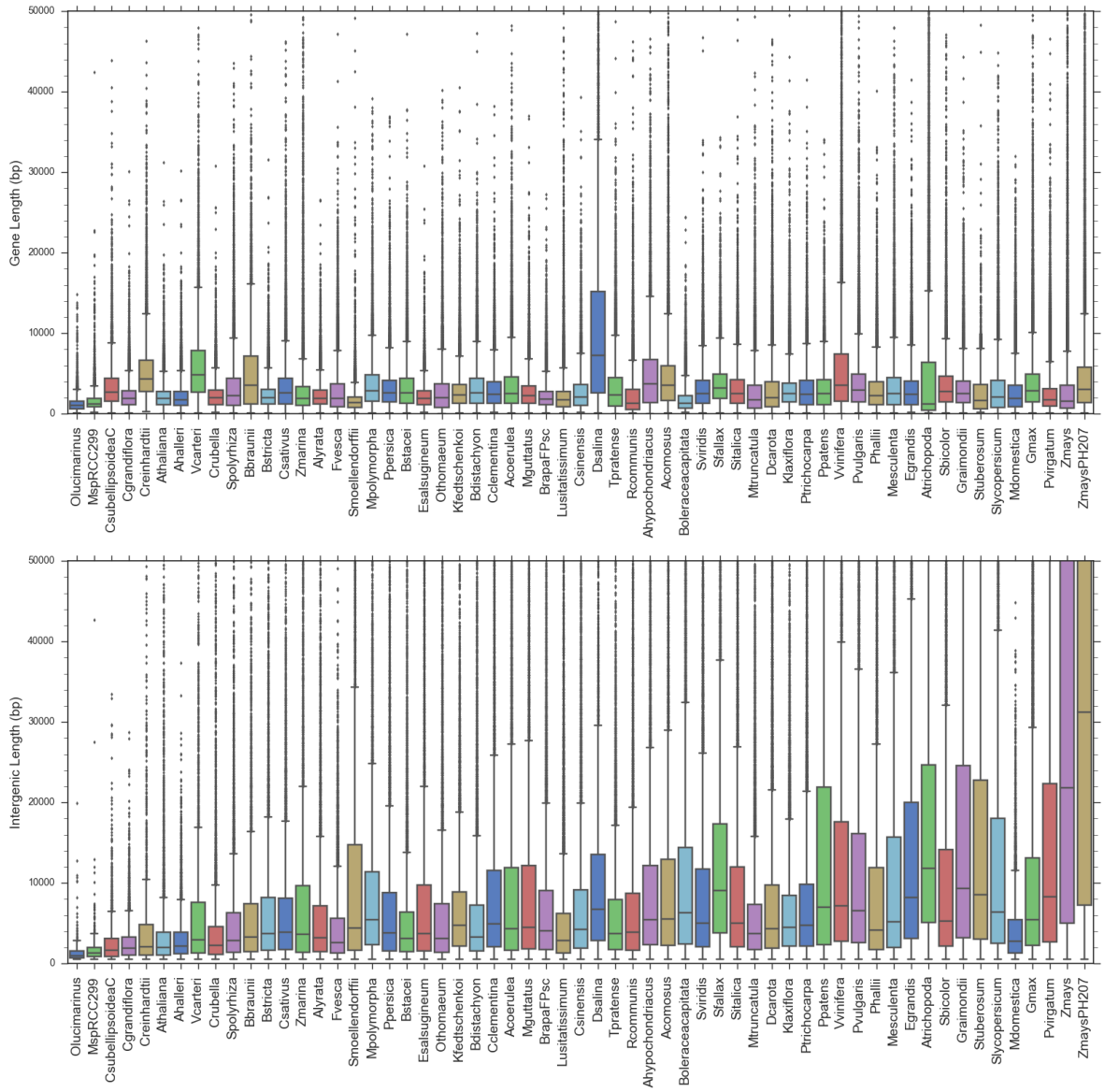


Figure 50. Expansion of genic and intergenic regions. The species are sorted from smallest genome (left) to largest (right). Gene length is highly consistent across the Embryophyta, while the Chlorophyta show a greater degree of variation in gene length. In particular, the Chlorophyta show longer genes, especially *D. salina*. These data indicate that intergenic regions are responsible for increases in genome size.

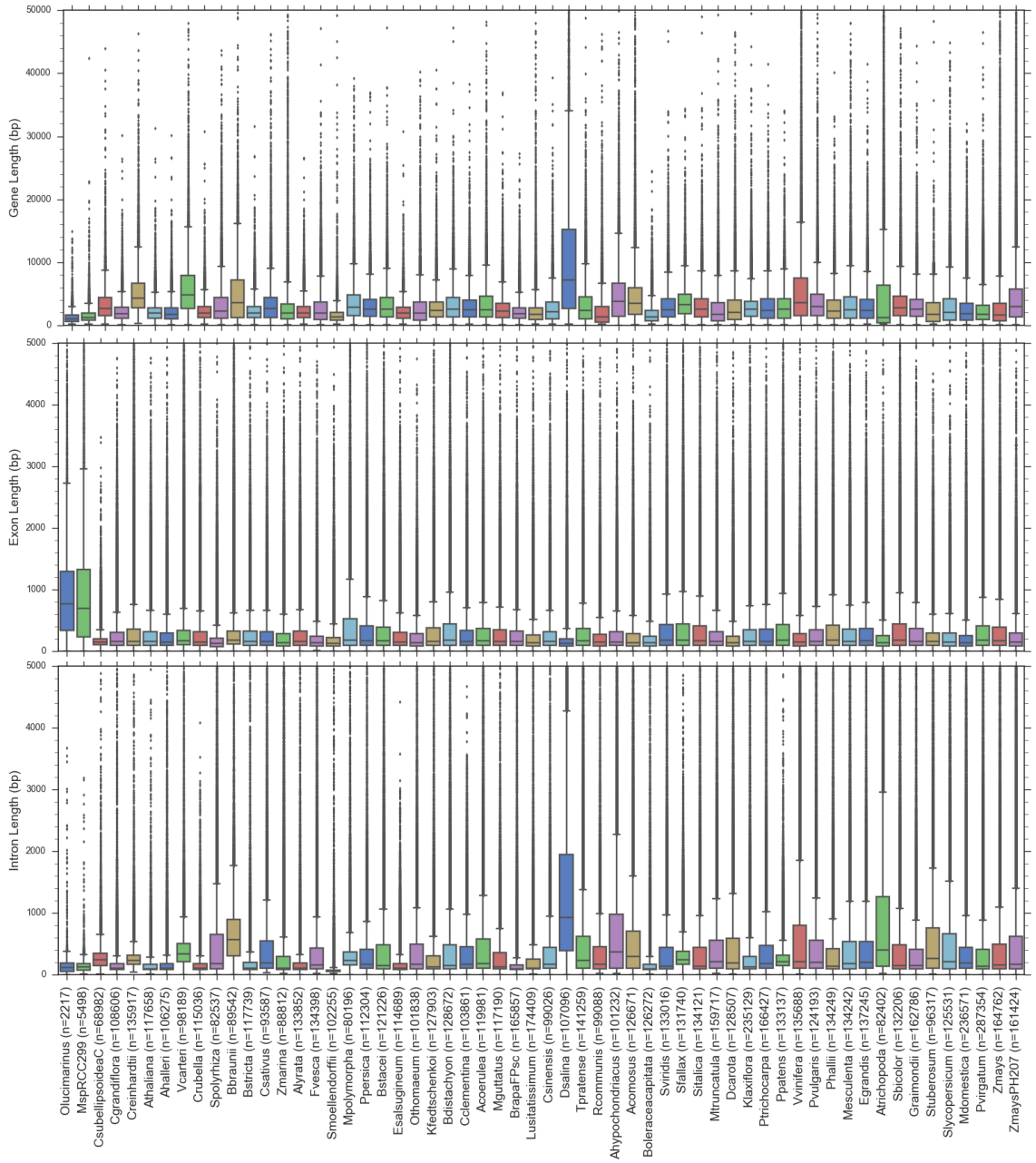


Figure 51. Introns drive expansion in gene length. These data show that exon lengths are highly consistent across the Viridiplantae, with the exceptions of *O. lucimarinus* and *M. pusilla*. However, these two species have very few introns. Intron lengths are more variable, and are largely responsible for variations in gene length.

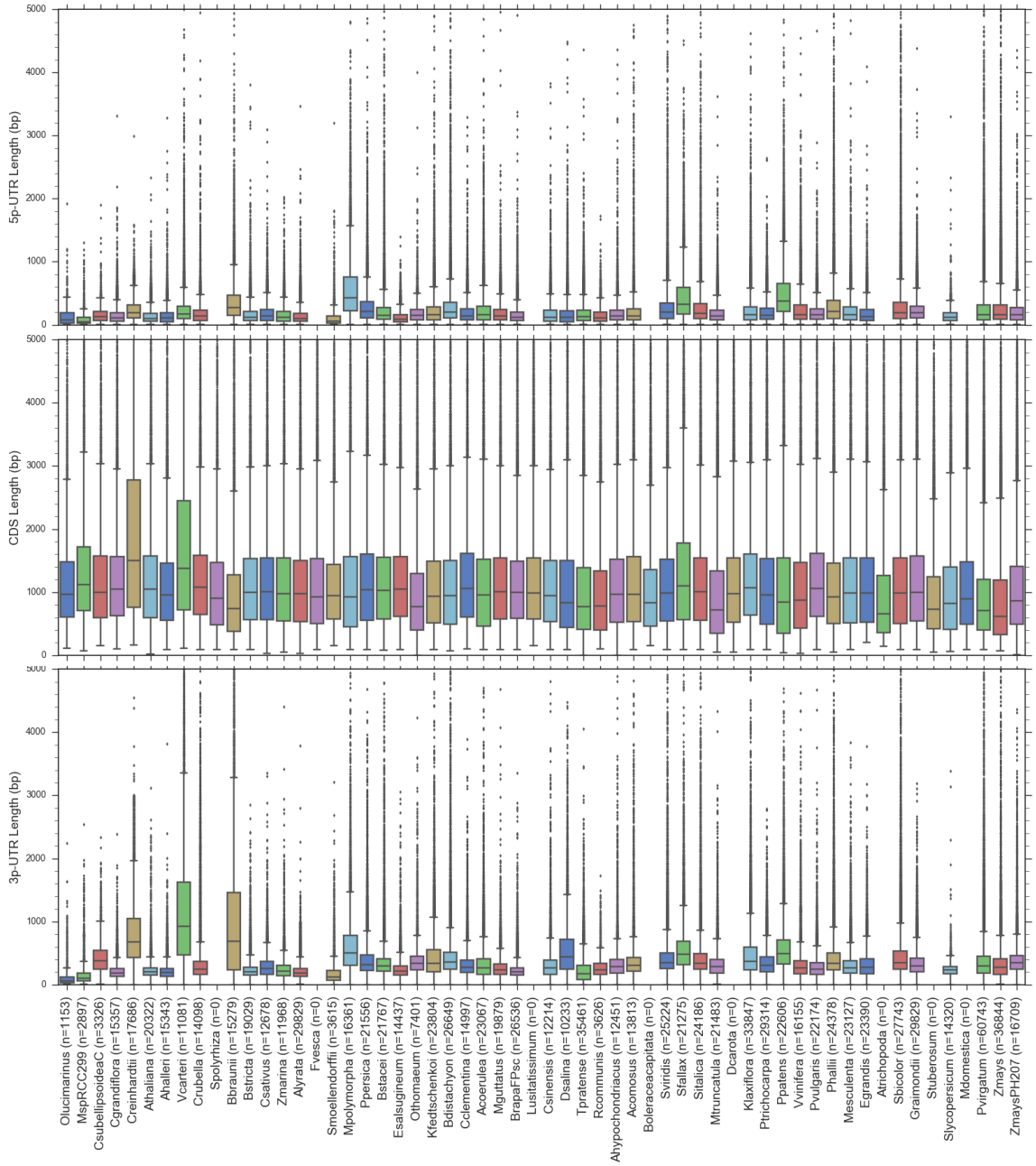


Figure 52. CDS length remains relatively constant. Gene structures are fairly consistent within the Embryophyta and more variable within the Chlorophyta. Across most of the Viridiplantae, CDS lengths are highly conserved, with the exceptions of *C. reinhardtii* and *V. carteri*, which have particularly long CDSs. While the 5'-UTR lengths are mostly consistent across species, the 3'-UTR lengths show greater variability.

3.3.3 Gene Evolution in Different Key Pathways

This section explores the annotations using the KEGG framework to analyze specific pathways of interest. The KEGG database has a useful API that is accessible over the internet and the BioPython package in particular allows for ease of access. Utilizing a custom script to access the KEGG API, process pathway information, and parse the databases presented in this work, specific pathways were analyzed. The tables of data generated with the custom Python code were then visualized with Morpheus and are discussed in detail below.

3.3.3.1 Protein Synthesis and Degradation

Looking at the aminoacyl-tRNA biosynthesis, there are two genes that are found only in Chlorophyta (Figure 36). The KEGG orthology term (K10837) is defined as an O-phosphoseryl-tRNA(Sec) kinase (PSTK, EC 2.7.1.164). The second term (K03341) is defined as an O-phospho-L-seryl-tRNASec:L-selenocysteinyl-tRNA synthase (SEPSECS, EC 2.9.1.2). In the biosynthesis of selenocysteinyl-tRNA (Sec-tRNA), the cognate tRNA is charged with a seryl moiety, which is phosphorylated (PSTK) and then converted to selenocysteine (SEPSECS) (*287*). Notably, the PSTK gene is missing in *B. braunii*, but it is possible that this is due to incompleteness of the genome. This demonstrates that the Chlorophyta are distinguished from the Embryophyta in their ability to synthesize selenocysteine. Otherwise, the complement of aminoacyl-tRNA sythetases is very complete across the Viridiplantae, with only a handful of missing annotations.

The ribosome is responsible for converting aminoacyl-tRNAs into polypeptide chains, and genes encoding the various ribosomal subunits show significant expansions in Embryophyta, compared to Chlorophyta (Figure 37). Interestingly, there are two ribosomal subunits *B. braunii* shares with Embryophyta but not Chlorophyta. One of them is the small subunit ribosomal protein

S3 (K02982), which lines entry to the ribosomal tunnel and plays an important role in mRNA helicase activity (*288*). There are two homologs in *A. thaliana*, a chloroplast-encoded protein (ATCG00800) (*289*), and a mitochondrial-encoded protein (ATMG00090) (*290*). The second one is the small subunit ribosomal protein S19 (K02965), which plays a role in conformational rearrangements during assembly of the small (30S) ribosomal subunit (*291*). There is one homolog in *A. thaliana* (AT5G47320), a nuclear-encoded mitochondrial ribosome subunit (*292*). There are two ribosomal proteins which are present in all Embryophytes but missing in all Chlorophytes (Figure 37). One is the large subunit ribosomal protein L14 (K02874), which is involved in controlling the relative movement of ribosomal subunits and inter-subunit bridges during translation (*293*). The yeast homolog is targeted by a silencing factor to inhibit translation (*294*). The other is the large subunit ribosomal protein L5 (K02931), an essential component in *E.coli* for the formation of the central protuberance during ribosome assembly (*295*). These data point to distinctive mechanisms of ribosome function between the Chlorophyta and the Embryophyta.

Polypeptides produced by ribosomes are degraded by the proteasome, one of the most highly conserved pathways in the Viridiplantae (Figure 38). A significant number of duplications have occurred in only a small subset of the species and only a few genes are missing across the board, most likely due to genome incompleteness. In *B. braunii*, there are three missing proteasomal genes. They are a non-ATPase structural component (K03033) in the lid of the 26S proteasome (*296*), a component (K03062) essential for channel opening of the proteasomal core particle (*297*), and a part of the core particle (K02725), which cleaves peptides bonds (*296*), and binds lipopolysaccharide (*298*). The proteasome recognizes proteins for degradation by detection of ubiquitin markers. The ubiquitin-mediated proteolysis system shows substantial expansions in Embryophyta, compared to Chlorophyta (Figure 39).

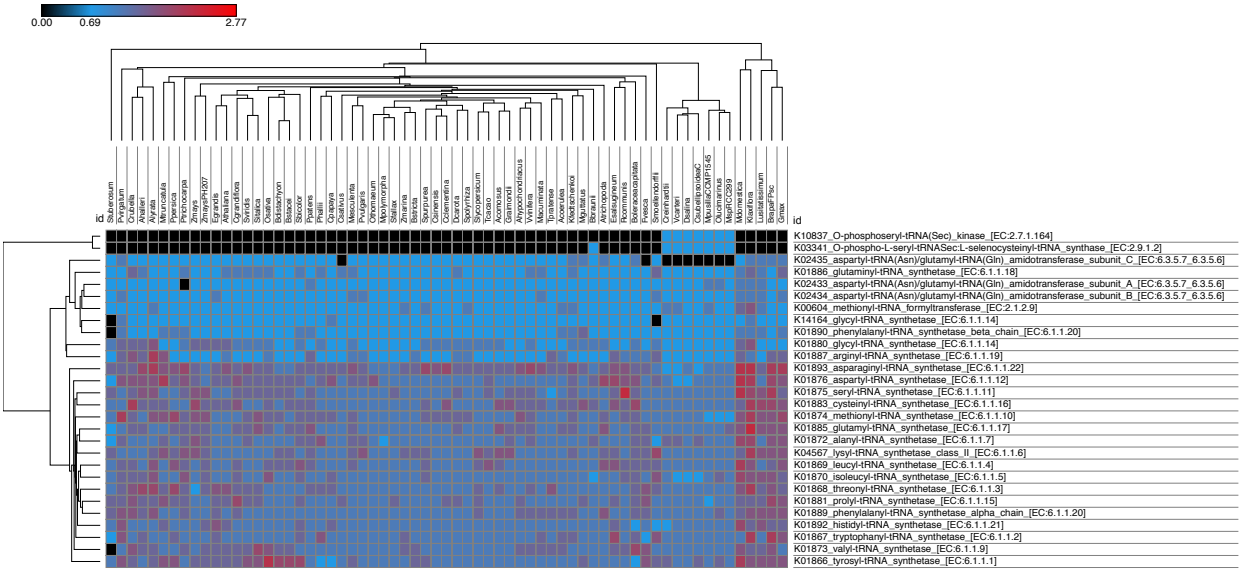


Figure 53. Aminoacyl-tRNA biosynthesis pathway. This pathway shows a high degree of conservation across all Viridiplantae. Interestingly, the Chlorophyta are distinguished from the Embryophyta by the presence of genes for biosynthesis of selenocysteine. Otherwise, missing genes are most likely the result of genome incompleteness.

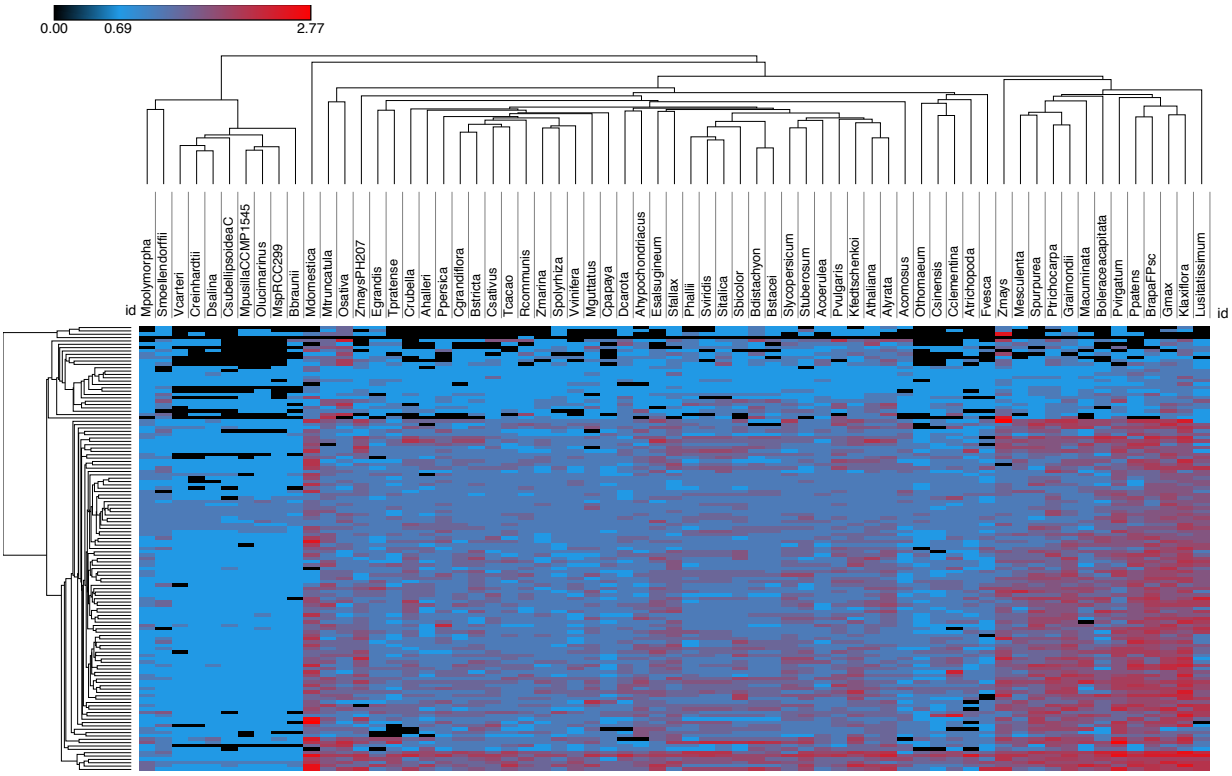


Figure 54. Proteins involved in ribosome assembly. Proteins in the ribosome assembly pathway are highly conserved, although the Embryophyta are distinguished by substantial increases in copy number of most genes in the pathway. In the Chlorophyta, the pathway consists largely of single-copy genes.

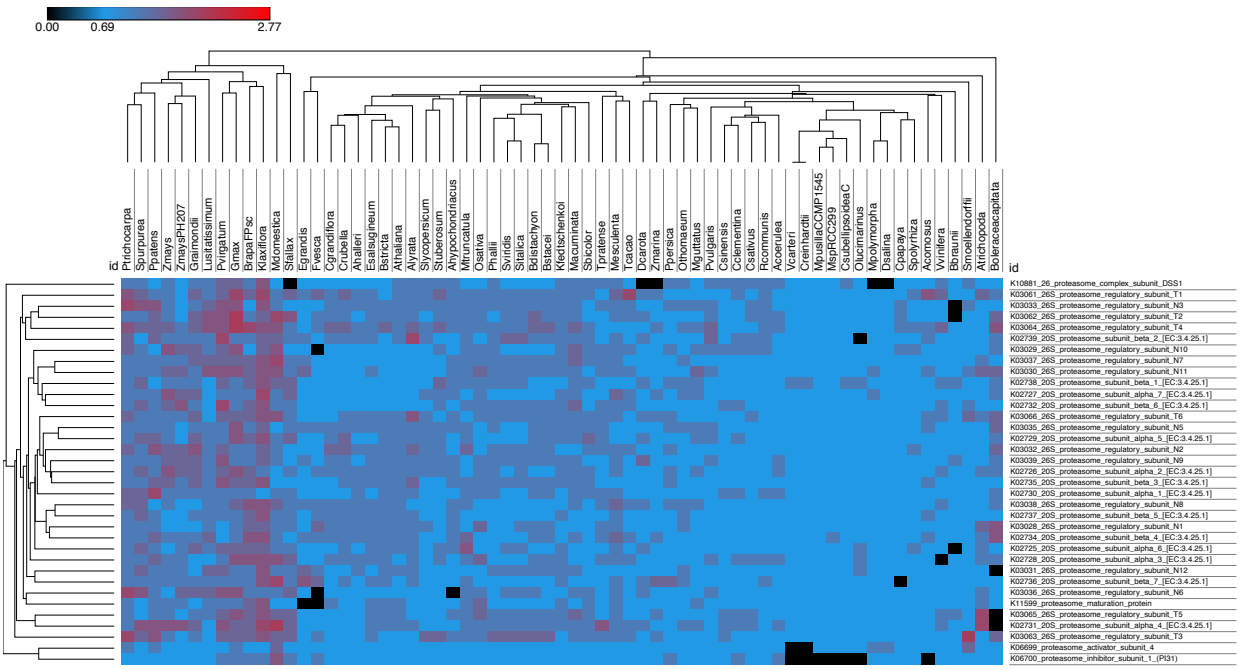


Figure 55. Proteins involved in proteasomal degradation. The proteasomal degradation pathway is perhaps the most conserved, with nearly all species containing the complete pathway. However, there are variations in copy-number of genes within the pathway among the species. Yet these copy-number variations do not appear to correlate well with traditional phylogenetic relationships.

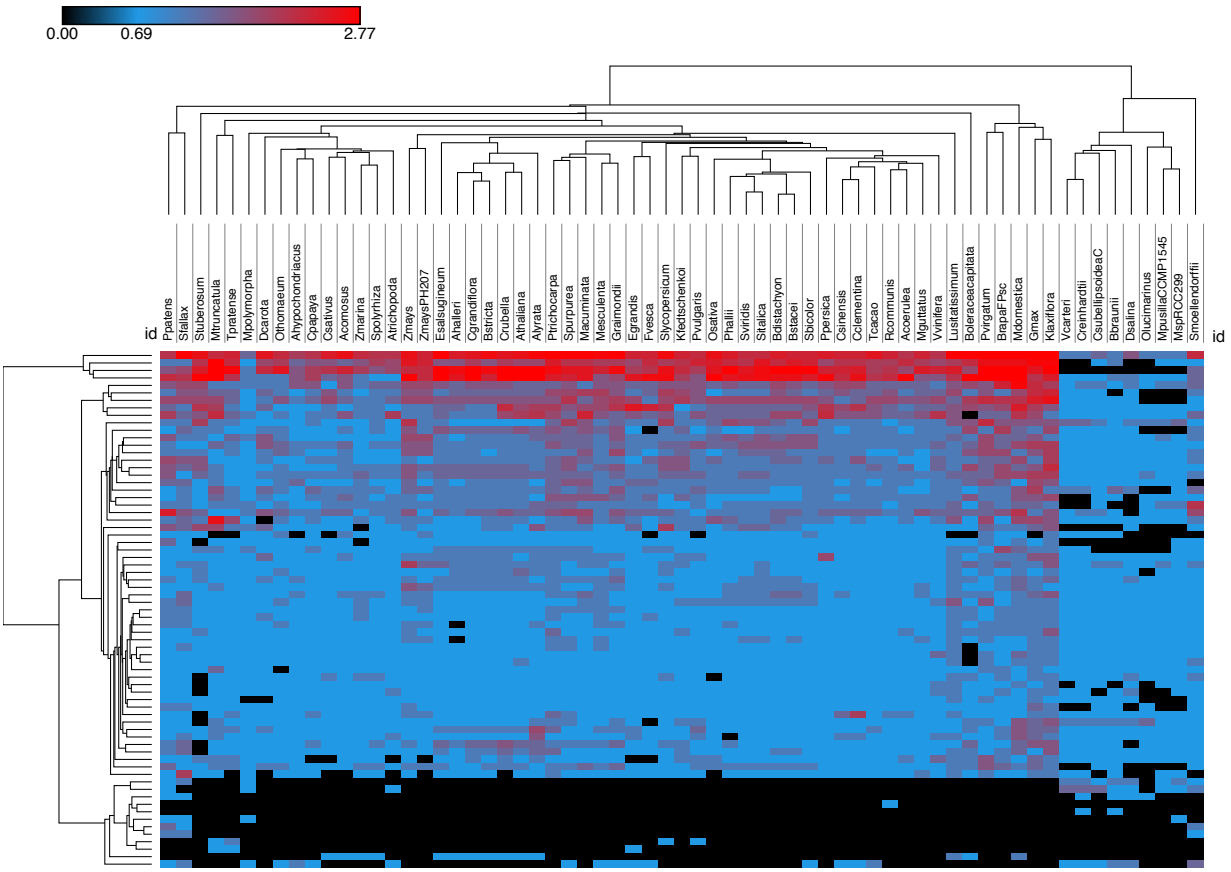


Figure 56. Proteins involved in ubiquitin-mediated proteolysis. The Chlorophyta are clearly dinstinguished from the Embryophyta in the ubiquitin-mediated proteolysis pathway. In the Embryophyta, a small sub-set of the genes in this pathway are highly enriched in copy-number. However, the Chlorophyta still have all the core components of the pathway.

3.3.3.2 Core Transcriptional Machinery

RNA polymerase (Pol) is essential for the ability to transcribe genes and shows a fairly high degree of conservation across Viridiplantae (Figure 40). There is one RNA Pol subunit missing from all Chlorophyta and only present in some Embryophyta. This is the bacterial-type DNA-directed RNA Pol subunit alpha (K03040), which is involved in transcription activation by cAMP-CRP in *E. coli* (*299*). There is a homolog in the *A. thaliana* chloroplast (ATCG00740) that undergoes RNA editing, disruption of which leads to significantly impaired gene expression (*300*), and is essential for plastid development (*301*). There are two subunits that are missing in *B. braunii* and some other species as well. These are the DNA-directed RNA Pol III subunits RPC5 (K14721) and RPC7 (K03024). RNA polymerase III serves as a cytosolic DNA sensor in mammalian cells, mediating innate immune responses to foreign DNA (*302*). In yeast, RNA polymerase III makes all tRNAs, the 5S rRNA, and various short, non-coding RNAs, which altogether account for around 15% of total transcription (*303*). RPC5 is involved in transcription termination and reinitiation (*304*), while RPC7 does not have an identified counterpart in RNA polymerase I or II (*305*). All of the RNA Pols are dependent on basal transcription factors, which also show a reasonable degree of conservation across Viridiplantae, with some types undergoing greater degrees of duplication (Figure 41). In the Chlorophyta, there is not much duplication of the basal transcription factors. However, *B. braunii* has 3 copies of transcription initiation factor TFIID subunit 6 (K03131), which is associated with the TATA-box binding protein (TBP) (*306*). The TBP is responsible for recognizing DNA promoter elements that drive transcription.

Processing RNA transcripts from RNA Pol is the responsibility of the spliceosome complex, which is largely conserved across Viridiplantae, with a small subset of factors undergoing duplications in Embryophyta (Figure 42). One term is missing from all Chlorophyta

but present in most Embryophyta. This is the U1 small nuclear ribonucleoprotein A (K11091), which can interact with a component of the polyadenylation complex, linking it with the spliceosome (*307*). *B. braunii* and *C. subellipsoidea* are the only two Chlorophytes that share with the Embryophytes a protein called apoptotic chromatin condensation induced in the nucleus (ACINUS) (K12875). In mammalians, ACINUS is targeted for cleavage by caspase-3 to induce apoptotic chromatin condensation without inducing DNA fragmentation (*308*), and also serves as part of the exon junction complex on mature mRNAs (*309*). Global to all species of Viridiplantae, but particularly frequent in Embryophyta, are the heat shock 70 kDa protein (hsp70) 1/2/6/8 (K03283) and heterogeneous nuclear ribonucleoprotein (hnRNP) A1/A3 (K12741). Hsp70 proteins perform a variety of functions in the cell, serving as chaperones and folding catalysts for many different processes (*310*). In mammalians, hnRNPs are involved in cytoplasmic mRNA trafficking, a critical function for directing cellular organization (*311*). There are two spliceosomal proteins missing only in *B. braunii* among the Viridiplantae. One is the PHD finger-like domain-containing protein 5A (K12834), that forms a part of the U2 small nuclear RNP complex, an important part of the spliceosome complex (*312*). The other is the THO complex subunit 3 (K12880), part of the transcription export (TREX) complex that regulates transport of RNA to outside of the nucleus (*313*). Interestingly, *B. braunii* has five copies of an ATP-dependent RNA helicase DHX8/PRP22 (K12818), which in humans is involved in nuclear export of spliced mRNA by releasing the RNA from the spliceosome (*314*).

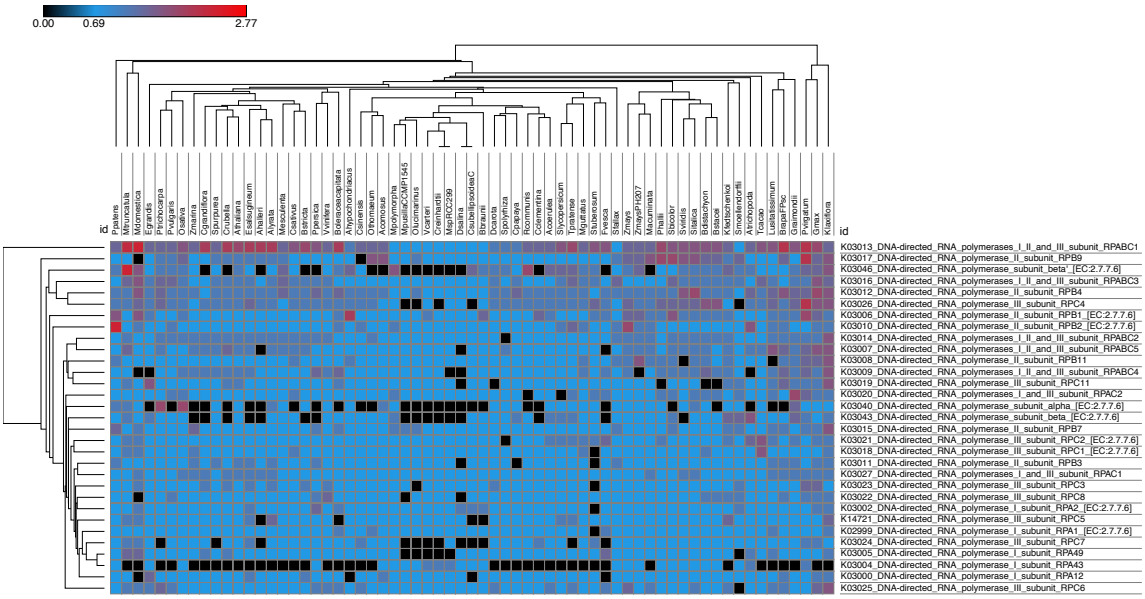


Figure 57. Protein components of RNA polymerase. These data show that transcription machinery proteins are highly conserved across the Viridiplantae. There is some variation in copy-number of the genes in this pathway, but there are also a number of missing genes. It is unclear whether the missing genes are genuinely absent or missing due to genome incompleteness.

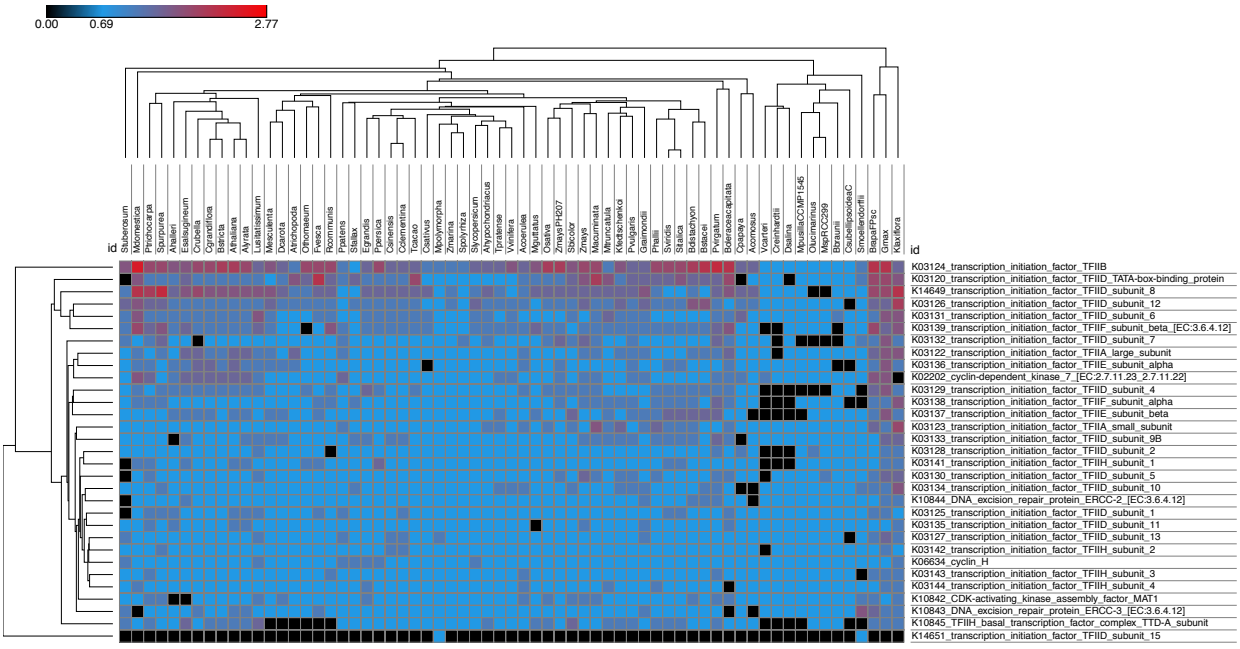


Figure 58. Basal transcription factors. These data show a high degree of conservation in the basal transcription factors across the Viridiplantae. There are a small number of genes missing from species in the Chlorophyta. Whereas in the Embryophyta there are many genes that have undergone at least one duplication.

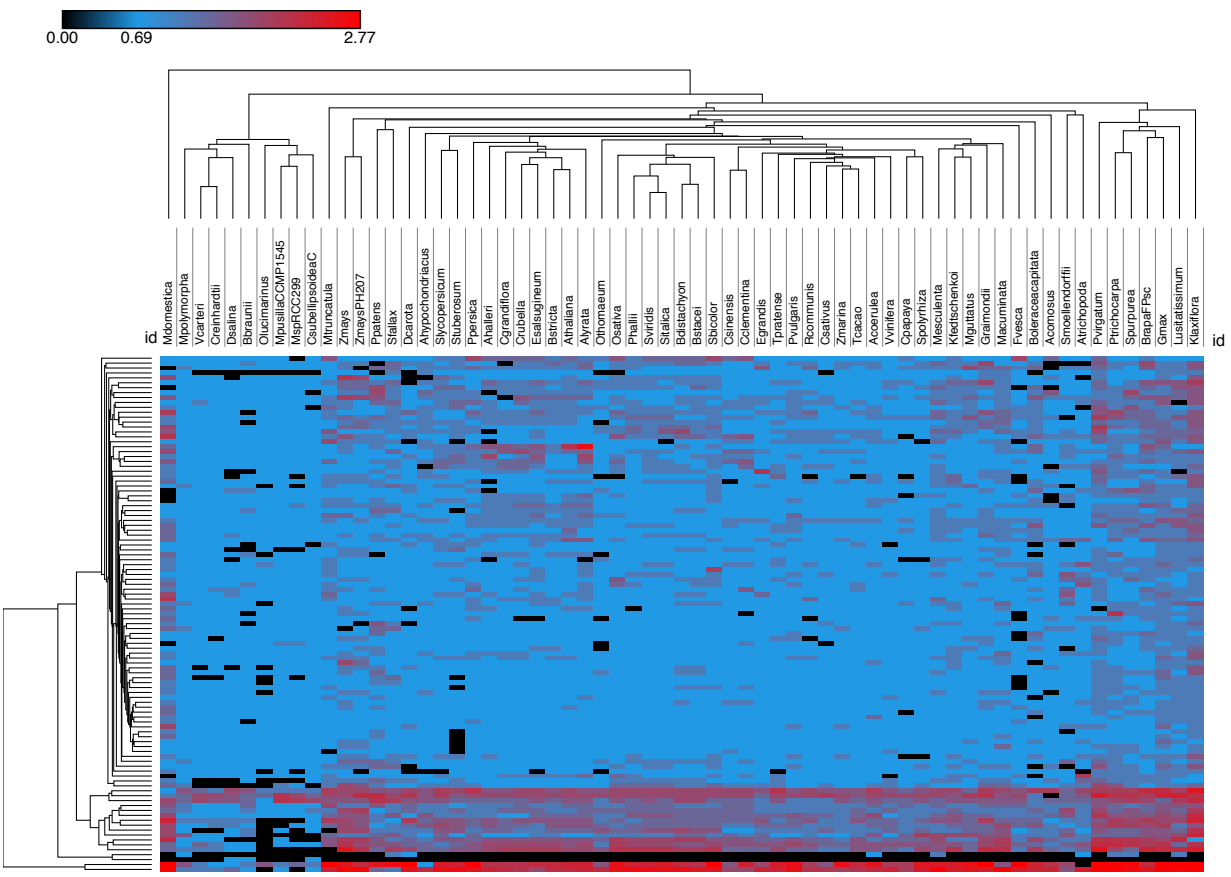


Figure 59. Protein components of the spliceosome. The genes in this pathway show a high degree of conservation, with a small sub-group of genes being highly duplicated in the Embryophyta. Otherwise the patterns of conservation are consistent across the Viridiplantae.

3.3.3.3 DNA Replication and Cell Division

The ability to replicate DNA is among the most basic definitions of life. That is, the ability to store genetic information over time. As expected, the complement of DNA replication proteins in Viridiplantae is highly conserved across all species (Figure 43). Surprisingly, only one protein showed a substantial amount of duplication, especially in the Embryophyta. The replication factor A1 (K07466) binds to single-stranded DNA (ssDNA), playing roles in DNA replication, repair, and recombination, actively coordinating these processes, and is essential for genome duplication and stability (*315*). There are two proteins that are global in Viridiplantae and have at least two copies in all Chlorophyta. They are replication factor C (RFC) subunit 2/4 (K10755) and subunit 3/5 (K10756). RFC is an ATP-dependent complex that loads proteins onto DNA for replication, repair, and modification (*316*). Proteins involved in homologous recombination show a similar degree of conservation to the DNA replication proteins, in part because of some overlap between the two sets (Figure 44). There are three genes that have a single copy present in all Chlorophyta, the DNA polymerase delta (POLD) subunit 2 (K02328) and subunit 4 (K03505), and the double-strand break repair protein MRE11 (K10865). POLD is thought to play a central role in the maturation of Okazaki fragments during DNA replication (*317*). MRE11 is part of a complex that integrates DNA repair with the activation of checkpoint signaling and is essential for double-strand break repair (*318*). There is one gene that has two or more copies in all Chlorophyta, the bloom syndrome protein (K10901), which is a helicase that is necessary for normal DNA double-strand break repair (*319*). Missing from all Chlorophyta, but present in Embryophyta, are the ATP-dependent DNA helicase RecG (K03655), a single-strand DNA-binding protein (K03111), and the crossover junction endonuclease EME1 (K10882). These proteins function in unwinding, stabilizing, and modifying DNA, respectively.

DNA replication and repair processes are only one function within the broader context of the cell cycle, which is highly conserved and show about a dozen proteins that have undergone significant duplications in Embryophyta (Figure 45). One gene is missing from all Chlorophyta, but present in most Embryophyta, the cell division control protein 7 (Cdc7) (K02214). Cdc7 is a kinase that plays a role in activating chromatin for assembly of the pre-replication complex, which ideally happens only once per cell cycle (*320*). There are three genes that are global in Viridiplantae and highly duplicated in Embryophyta. The S-phase kinase-associated protein 1 (K03094) plays a role in ubiquitin-mediated proteolysis of proteins with an F-box domain, such as cyclin A-CDK complexes (*321*). In mammalians, cyclin A proteins (K06627) play a role in preventing centrosome reduplicatin (*322*), thereby maintaining genome stability, and may also play a role in cell motility (*323*). The G2/mitotic-specific cyclin-B1 protein (K05868) forms complexes with CDK1 that are responsible for restricting cell growth prior to cell division, an essential function of the cell cycle (*324*). Meiosis is another critical function of the cell cycle, especially with the development of more complex tissue types in higher plants (Figure 46). Some of the global Viridiplantae genes that are conserved at low copy number include the cell division control 45 (CDC45) protein (K06628), and the DNA replication licensing factors MCM3 (K02541) and MCM5 (K02209), the origin recognition complex (ORC) subunit 1 (K02603). CDC45 assembles into a complex with MCM5 that is essential for chromosomal DNA replication (*325*). MCM5 also interacts directly with cyclin A and indirectly with ORC in the process of preventing centrosome reduplication (*322*). The MCM complex is a heterohexamer with DNA helicase activity that functions in DNA replication and is loaded onto chromatin in a cell cycle-dependent manner (*326*). DNA replication is initiated by loading the ORC onto chromatin, which serves as a scaffold to assemble the remaining replication machinery (*327*).

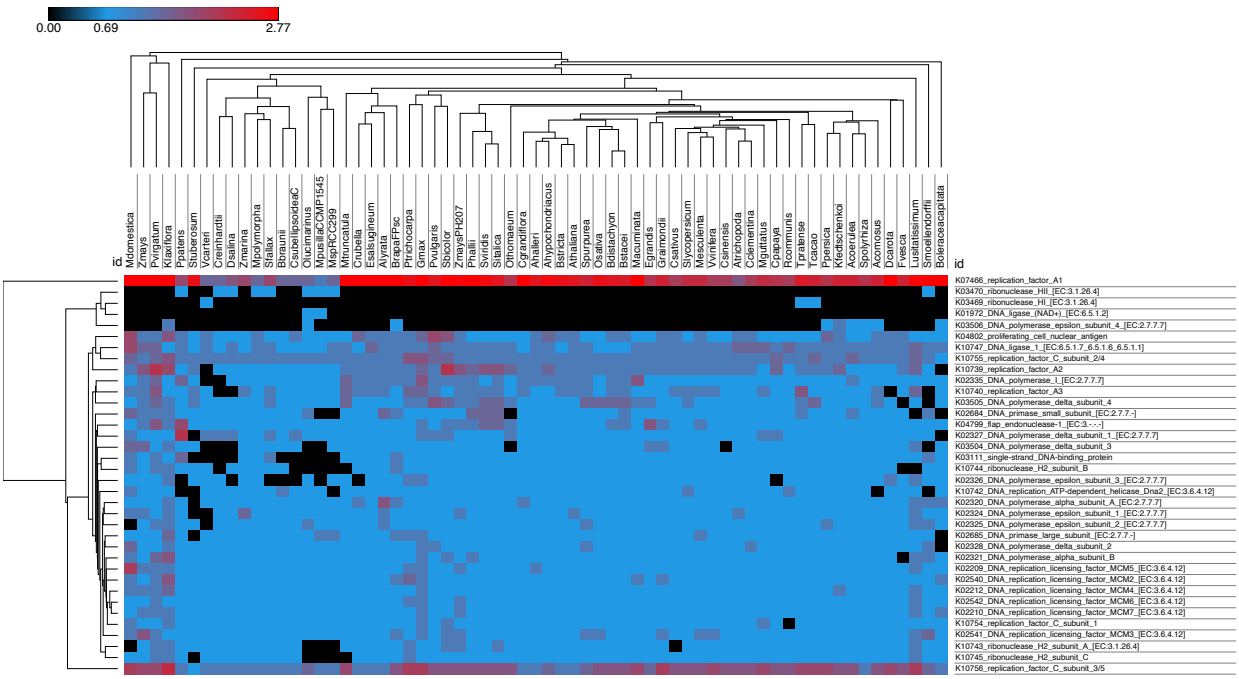


Figure 60. Proteins involved in DNA replication. There is only one gene in this pathway that has been highly duplicated in the Embryophyta, otherwise the pathway is consistently conserved across the Viridiplantae. The dendrogram of species does not approximate the estimated phylogenetic relationships.

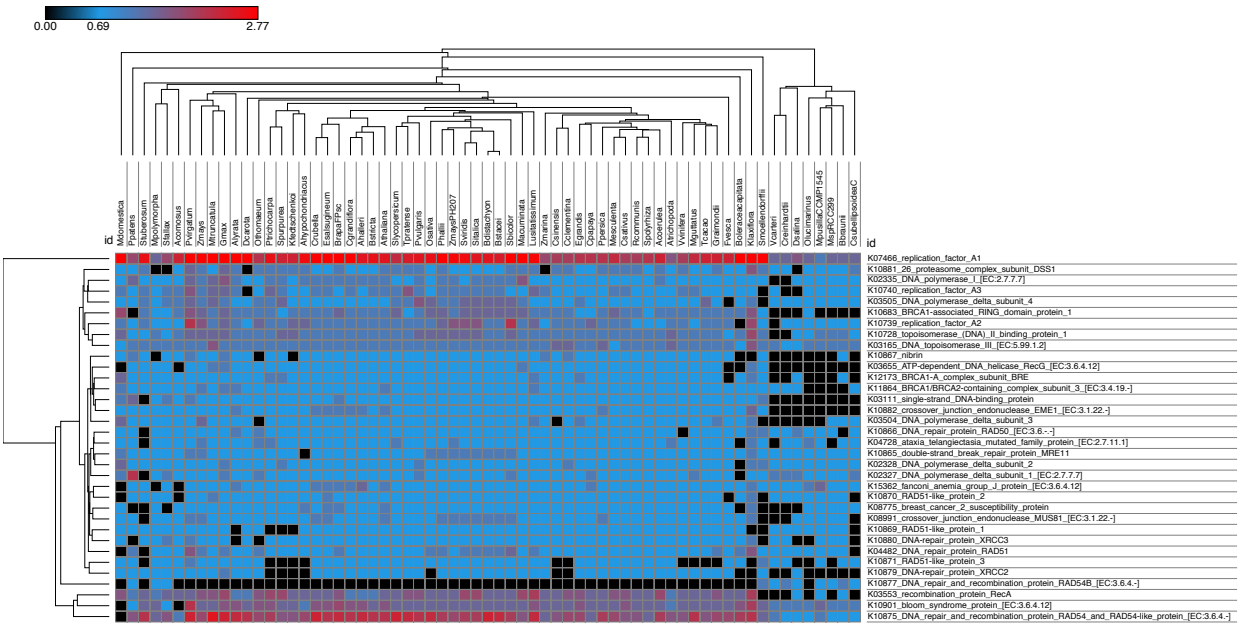


Figure 61. Proteins involved in homologous recombination. There is a large group of genes in this pathway that are missing in the Chlorophyta. In the Embryophyta, one gene has been highly duplicated, while a small number of genes have undergone a lesser number of duplications.

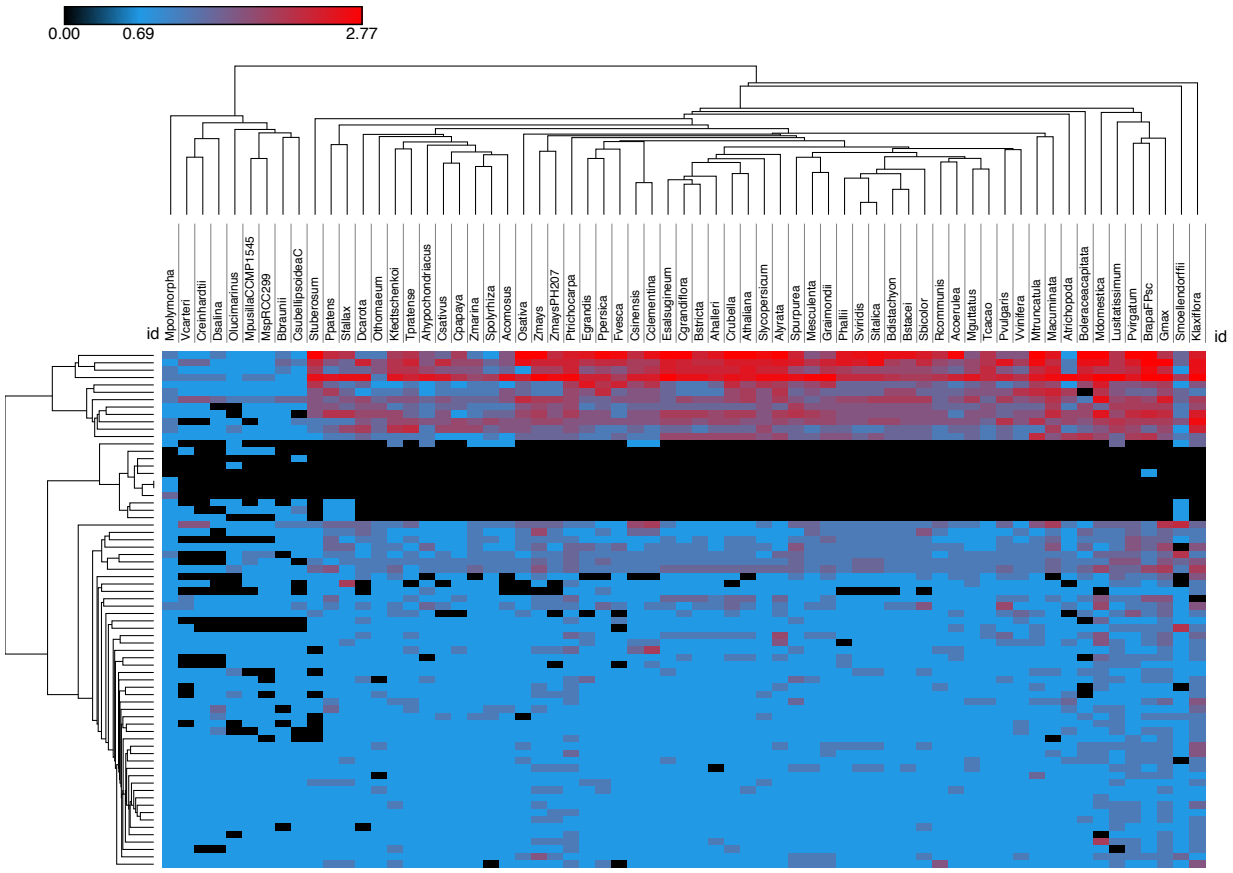


Figure 62. Proteins involved in the cell cycle. Most genes in this pathway are highly conserved, with nearly a dozen genes being highly duplicated in the Embryophyta. The Chlorophyta are only entirely missing one genes from the pathway, compared against the Embryophyta.

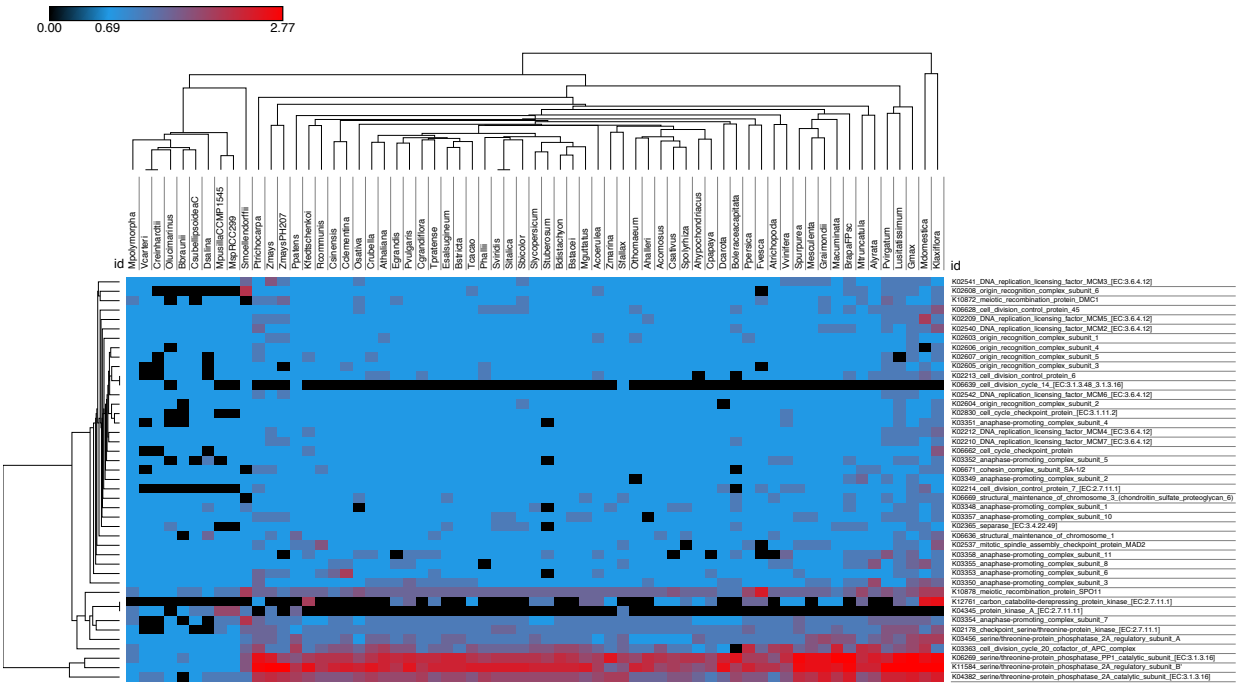


Figure 63. Proteins involved in meiosis. There are three genes that have undergone substantial duplication in the Embryophyta, but remain single- or double-copy in the Chlorophyta. There are factors present only in the Chlorophyta and only in the Embryophyta, as well as in both.

3.3.3.4 Photosynthesis and Carbon Fixation

The ability to produce energy via photosynthesis is a defining feature of Viridiplantae. The spread of genes involved in photosynthesis does not correspond well with the accepted phylogenetic tree of species in Viridiplantae (Figure 47). This suggests that evolution in the photosynthesis pathway is not a hallmark of speciation. Furthermore, while some genes are highly conserved, others appear to have evolved independently in multiple lineages. For example, there are a number of photosynthesis proteins that *B. braunii* shares with the Embryophyta, but not the Chlorophyta. Among these are the PSII proteins psbI (K02710), psbJ (K02711), psbL (K02713), psbT (K02718), and psbZ (K02724), all of which have homologs in the *A. thaliana* chloroplast (ATCG00080, ATCG00550, ATCG00560, ATCG00690, and ATCG00300, respectively). These proteins have various functions in the PSII complex, such as structural support for assembly and dimerization of the core particles, binding quinone and chlorophyll, and interacting with light-harvesting antennae (*328*). *B. braunii* also has copies of psaA (K02689), psaB (K02690), petB (K02635), petG (K02640), ATPF1B (K02112), ATPF0C (K02110), and ATPF0A (K02108), not found in the other Chlorophyta. The genes ATCG00350, ATCG00340, ATCG00720, ATCG00600, ATCG00480, ATCG00140, and ATCG00150 encode the respective homologs in the *A. thaliana* chloroplast. That none of these proteins would be annotated in the other Chlorophyta is quite surprising, given the importance of some, such as petB and petG, which are essential for the cytochrome b6f complex. Moreover, *C. reinhardtii* is known to have a copy of petG, deletion of which disrupts the cytochrome b6f complex (*329*). Either there are choroplastic sequences included in the nuclear genome assemblies of some species, or there is some other source of inconsistency among the photosynthesis annotations.

Putting aside these uncertainties, there are some interesting patterns in the presence of genes related to porphyrin and chorophyll metabolism (Figure 48). Missing from all Chlorophyta, but present in most Embryophyta are the red chlorophyll catabolite reductase (RCCR) (K13545), phytochromobilin:ferredoxin oxidoreductase (HY2) (K08101), and chlorophyllase (K08099). In *A. thaliana*, the enzymatic activity of RCCR (AT4G37000) results in the addition of a double bond to the porphyrin ring of chorophyll during the process of breakdown, but the gene also appears to play a role in mediating the cell death response to pathogens (*330*). HY2 is responsible for a step in biosynthesis of the light-harvesting prosthetic group of phytochrome photoreceptors, but in fact represents just one reaction in a broader family of closely related enzymes (*331*). Chlorophyllase is the first enzyme in the chlorophyll degradation pathway, and in *A. thaliana* its activity is enhanced by the presence of methyl jasmonate (*332*). Conversely to the above genes, there are two genes found in Chlorophyta but not Embryophyta. These are cob(I)alamin adenosyltransferase (K00798) and cytochrome c heme-lyase (K01764). While there is some clade-specific signatures in the pathways discussed above, the carbon fixation pathway is nearly universal in the Viridiplantae species (Figure 49). The only major feature that distinguishes the carbon fixation pathway of the Embryophyta from the Chlorophyta is the duplication of certain genes. Among these are malate dehydrogenase (K00026), fructose-bisphosphate aldolase class I (K01623), malate dehydrogenase (oxaloacetate-decarboxylating) (K00029), phosphenolpyruvate carboxylase (K01595), glyceraldehyde 3-phosphate dehydrogenase (K00134), and ribose 5-phosphate isomerase A (K01807). These enzymes participate in multiple pathways, such as the Calvin-Benson cycle, the pentose phosphate pathway, the citric acid cycle, and possibly others. Variable expression and localization of these enzymes could lead to a wide array of different metabolic outcomes.

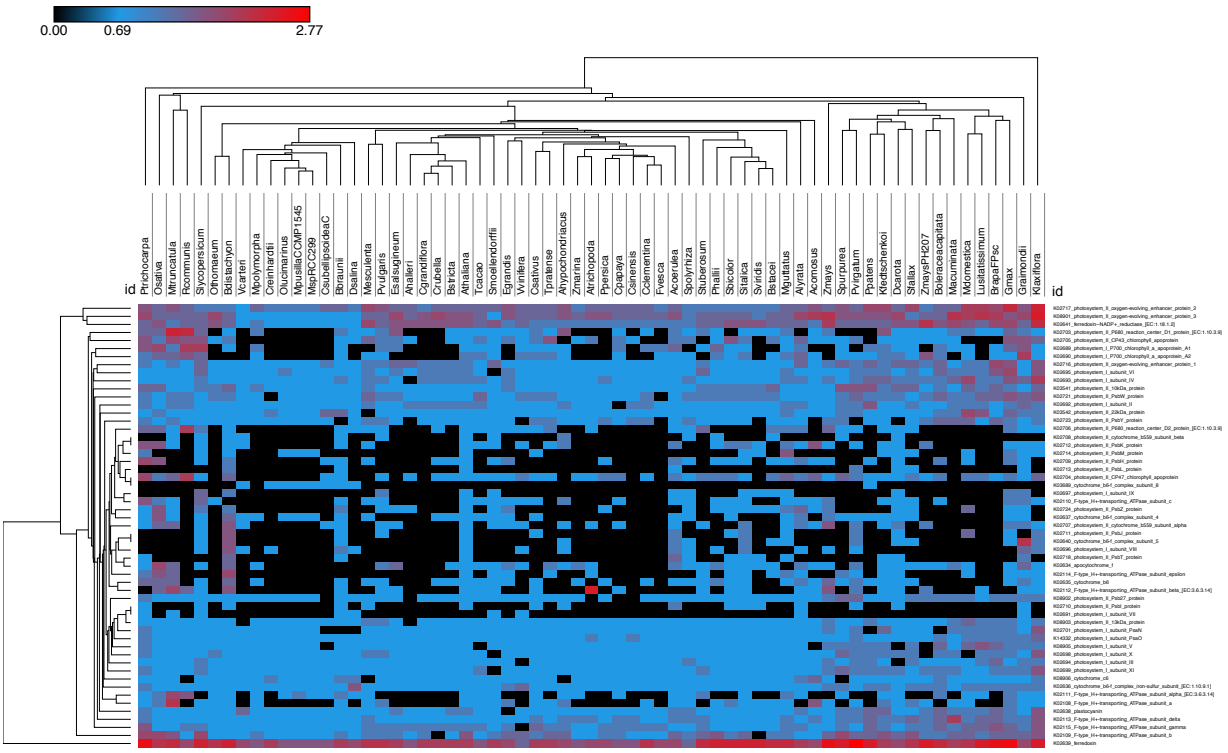


Figure 64. Proteins involved in photosynthesis. This pathway shows the most variation in the pathways that were analyzed. There appears to be a large amount of convergent evolution in photosynthesis across the Viridiplantae. However, it is difficult to separate true chloroplast-located sequences from endosymbiotic gene transfers. Variable methods of genome assembly could contribute to the apparent discrepancy of the genes in this pathway.

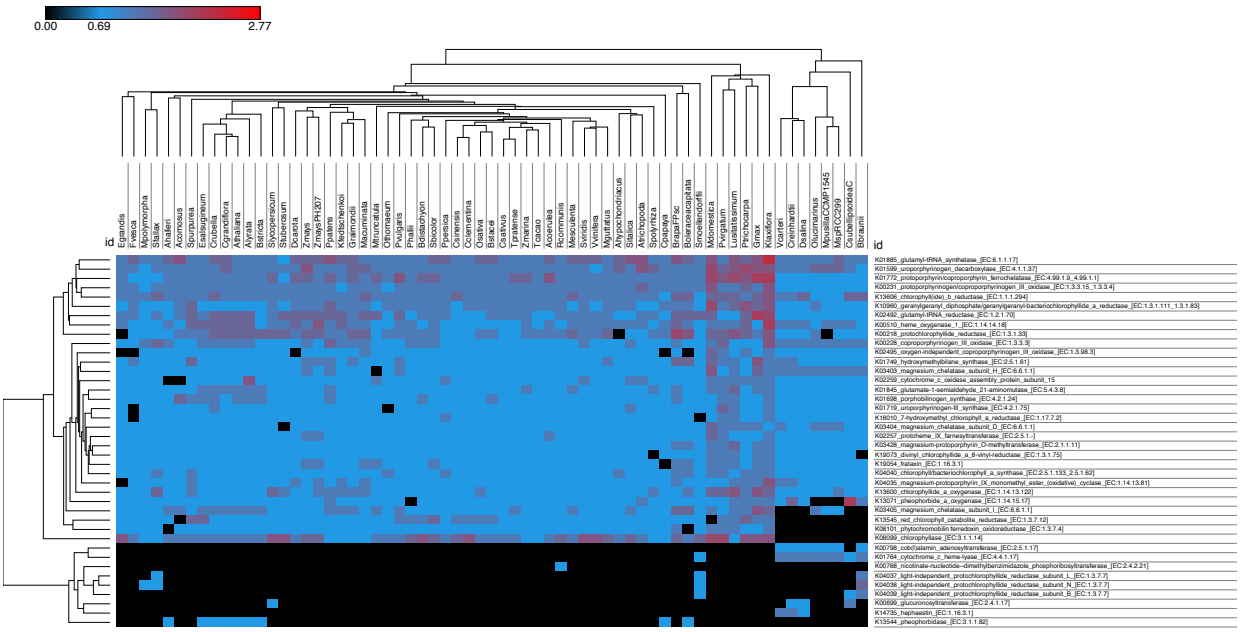


Figure 65. Proteins involved in porphyrin and chlorophyll metabolism. This pathway is highly conserved across the Viridiplantae. There are two genes present in the Embryophyta that are missing in the Chlorophyta, and vice versa. The Chlorophyta are distinguished by the presence of cytochrome c-heme lyase and cobalamin adenosyltransferase.

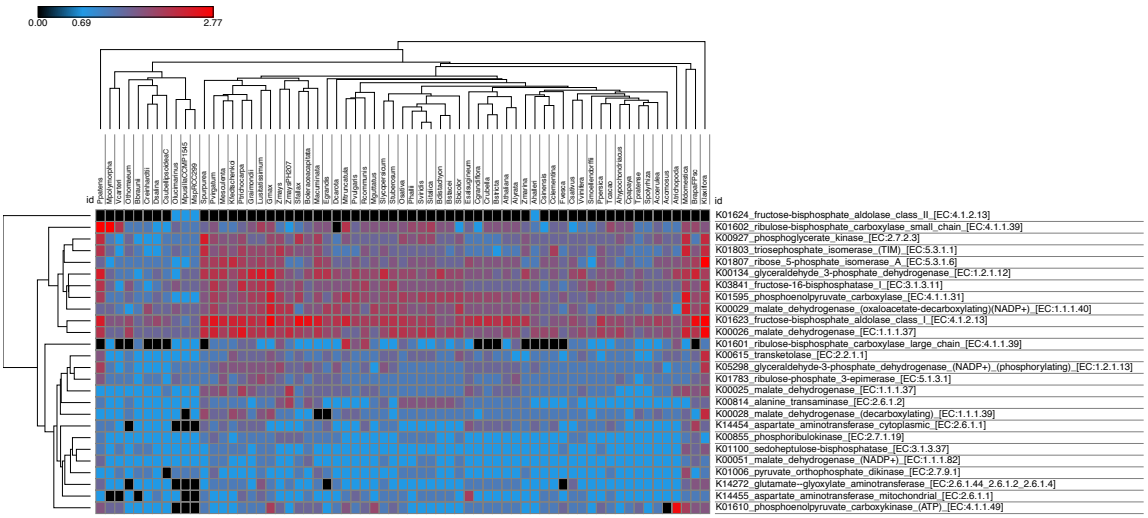


Figure 66. Proteins involved in carbon fixation. This pathway shows a large amount of positive selection on nearly half of the genes for species of the Embryophyta, compared against Chlorophyta. Across the Viridiplantae, there are very few genes missing in this pathway.

3.3.3.5 Central Energy and Carbon Metabolism

With the energy and carbon from photosynthesis, cells construct more complex products, beginning with several primary pathways that yield substrates for secondary pathways. Although photosynthesis produces ATP, more can be created via oxidative phosphorylation, components of which are highly conserved in Viridiplantae (Figure 50). Some genes are highly conserved, while others are sparsely distributed. There are four genes missing in all Chlorophyta, but present in most Embryophyta. The NAD(P)H-quinone oxidoreductase, subunit 5 (K05577), subunit I (K05580), and subunit J (K05581), and the F-type H+-transporting ATPase subunit 8 (K02125). NAD(P)H-quinone oxidoreductase is better known as complex I of the mitochondrial respiratory electron transport chain (*333*). There are three genes that *B. braunii* shares with Embryophyta, but not Chlorophyta, the complex I chains 5 (K03883), 6 (K03884), and 4L (K03882). Two genes are global in Viridiplantae and highly duplicated in Embryophyta, inorganic pyrophosphatase (K01507) and H+-transporting ATPase (K01535). Inorganic pyrophosphatase is a well-studied enzyme that is essential for regulating mitochondrial membrane potential (*334*). Aside from photosynthesis, a major source of energy for oxidative phosphorylation comes from the citric acid cycle, which is very highly conserved among the Viridiplantae (Figure 51). Interestingly, there are three genes that are present only in Chlorophyta. These are pyruvate carboxylase (K01958), fumarate hydratase class I (K01676), and succinate dehydrogenase (SDH) membrane anchor subunit (K00237). The presence of pyruvate carboxylase in Chlorophyta is interesting because it could provide a mechanism of carbon assimilation that parallels RuBisCO, or also an alternative to the pyruvate dehydrogenase complex. The membrane anchor subunit suggests that the SDH complex is attached to a membrane.

With energy from the oxidative phosphorylation and carbon from photosynthesis, cells can synthesize essential molecules, such as fatty acids, a small and highly conserved biosynthetic pathway (Figure 52). There is one gene in the pathway missing from all Chlorophyta but present in almost all Embryophyta, the fatty acyl-ACP thioesterase B protein (K10781), which is the major determinant of fatty acid level and chain length in plants (*335*). There are four genes that are global in Viridiplantae and highly duplicated in Embryophyta. They encode the proteins long-chain acyl-CoA synthetase (LACS) (K01897), 3-oxoacyl-ACP synthase II (K09458), 3-oxoacyl-ACP reductase (K00059), and acyl-ACP desaturase (K03921). In *A. thaliana*, LACSs play a role in the development of pollen (*336*) and cuticle (*337*), and their expression in yeast facilitated the uptake of fatty acids (*338*). In addition to fatty acids, terpenoids are a large and important class of metabolites, with very highly conserved machinery in Viridiplantae species (Figure 53). Interestingly, there are six genes present in all Embryophyta but missing in all Chlorophyta. They encode the proteins hydroxymethylgluratryl-CoA reductase (K00021), mevalonate kinase (K00869), diphosphomevalonate decarboxylase (K01597), NAD+-dependent farnesol dehydrogenase (K15891), phosphomevalonate kinase (K00938), and farnesol kinase (K15892). It is very interesting that no farnesol kinase enzyme was predicted in *B. braunii*, because there is direct biochemical evidence of exactly this reaction (*339*). There are three genes which have undergone significant duplications in the Embryophyta. They are 1-deoxy-D-xylulose-5-phosphate synthase (K01662), geranylgeranyl diphosphate synthase type II (K13789), and ditrans,polycis-polyprenyl diphosphate synthase (K11778). The latter enzyme is implicated in the biosynthesis of rubber (*340*).

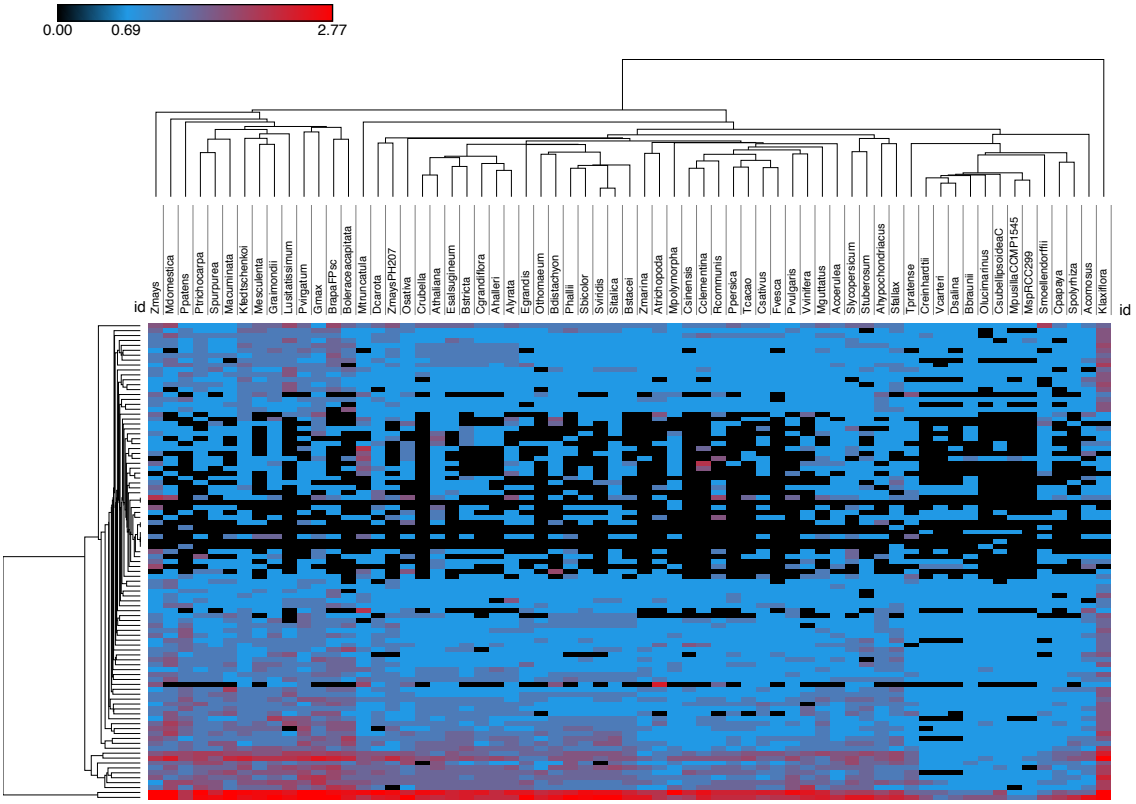


Figure 67. Proteins involved in oxidative phosphorylation. This pathway contains a large number of genes, with three sub-populations. There are highly conserved low-copy number genes, highly duplicated genes, and highly variable genes. These data indicate that oxidative phosphorylation is a highly complex pathway subjected to multiple evolutionary pressures.

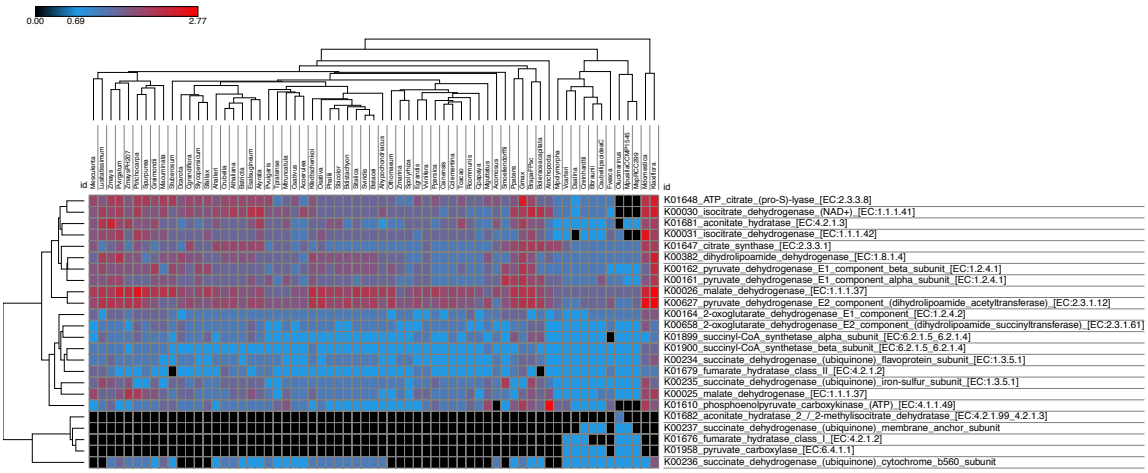


Figure 68. Proteins involved in the citric acid cycle. This pathway is highly conserved and enables the clear distinction of the Chlorophyta. They exclusively contain fumarate hydratase, pyruvate carboxylase, and succinate dehydrogenase. Nearly half of the genes have undergone highly positive selective pressures in the Embryophyta.

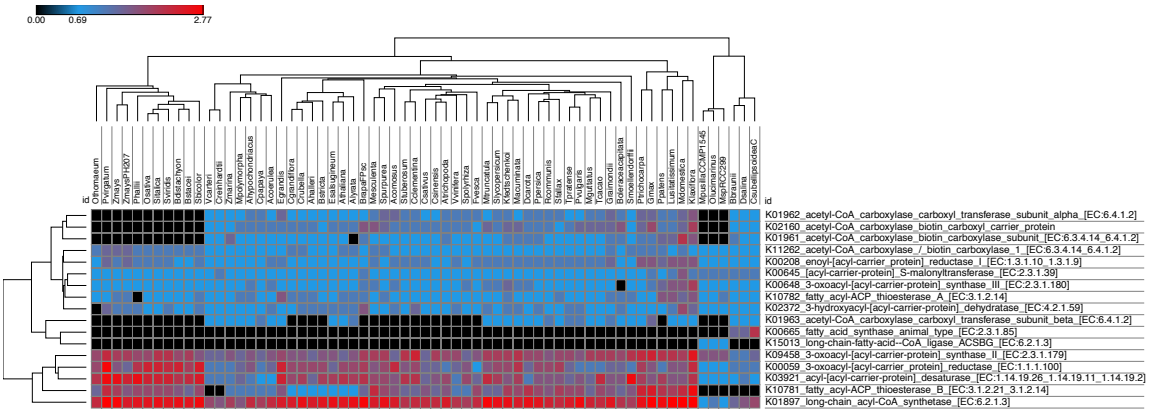


Figure 69. Proteins involved in fatty acid biosynthesis. This is a small, fundamental pathway that is highly conserved and shows positive, negative, and neutral selective pressures in distinct sub-sets of genes. The Chlorophyta and the Embrophyta are not well distinguished by hierarchical clustering.

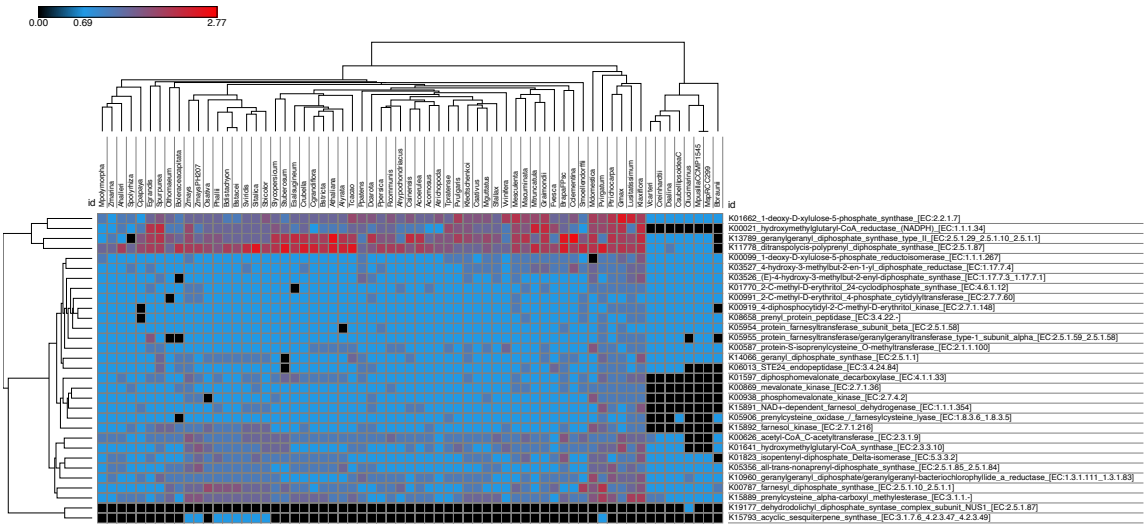


Figure 70. Proteins involved in terpenoid backbone biosynthesis. This pathway is highly conserved, mostly at low copy-number. There are four genes that have undergone highly positive selection in the Embryophyta. The Chlorophyta do not contain hydroxymethylglutaryl-CoA reductase, diphosphomevalonate kinase, mevalonate kinase, dophosphohomevalonate decarboxylase, and farnesol kinase.

3.4 Conclusion

This work has demonstrated a novel method for the genome-scale comparative analysis of Viridiplantae. The scripts developed in this work enable the easy transformation of data from Phytozome into tables for analysis. Future work should explore the application of different statistics to classify genes based on frequency of occurrence in the genomes. Furthermore, the evolutionary relatedness of different genomes becomes apparent through hierarchical clustering of the columns (species) in the table.

The functional tables proved immensely useful in combination with the KEGG pathway to further parse the gene contents into pathways. These sets of functionally related genes revealed distinctive patterns of evolution and selective pressures. Further applications could involve modeling metabolic pathways and reconstructing protein interactions networks. Perhaps 3-dimenensional whole-cell modeling of cellular compartments, metabolites, protein, and nucleic acids could one day become possible. This technology would enable the *de novo* design and construction of entirely new forms of life.

In addition to the functional analyses, for the first time, GenHub was applied to the Viridiplantae to reveal structural aspects of the genomes. As the contiguity of genome assemblies improves, gene organization analyses will become better. This will enable analysis of genome evolution in Viridiplantae over deep time. The reconstruction of ancestral genomes (karyotypes) could show the exact paths of evolution across lineages. This will help us better understand distinctive genomic features of given species of interest.

4. DIEL CYCLES IN *BOTRYOCOCCUS BRAUNII*

Genomics provides insight into the complete set of genes present in an organism, but systems biology also involves dynamic processes, such as transcription and metabolism. Therefore in order to obtain a more complete understanding of the holistic biological systems operating in *B. braunii*, experiments were designed and executed to obtain information about transcription and metabolism over time. In this section, the experimental design is explained, the data collection methods are described, the results are presented, and interpretations are discussed.

4.1 Introduction

The following section provides critical background information about the history and the state of the art in diel regulation of key biological processes, the experimental setup and its purpose, and the methods of biomass collection for downstream analyses. This information is essential context for understanding the resulting data.

4.1.1 Functional Regulation by Clocks and Cycles

The study of biological rhythms has a long and rich history involving the use of algae and plants. For example, in 1960, Melvin Calvin and members of his laboratory used synchronously grown *Chlorella* and ^{14}C-labeling (i.e. pulse-chase) to study metabolic changes throughout the cell cycle (*341*). They also elegantly summarized competing contemporary methods for growing synchronous cultures and the implications with respect to the data. One method involved synchronizing the cells by the intermittent application of light (i.e. diel cycles). In 1971, Surzycki (*342*) reported a very simple method to grow synchronous cultures of *C. reinhardtii*, by applying a 12-hour light, 12-hour dark diel cycle with constant temperature. In 1994, Krupinska and

Humbeck (*343*) comprehensively reviewed the achievements of cell cycle and circadian clock research in the 20th century, mainly using the tool of light-induced synchronous algae cultures. The community of scientists produced a body of work that offers broad fundamental insights into core mechanisms regulating organelle and cell division. Many studies investigated changes in DNA, RNA, and protein synthesis across the stages of the cell cycle, and its relationship with the circadian clock. By 2001, even greater knowledge of circadian rhythms in microalgae had accrued, as reviewed by Mittag (*344*). Circadian rhythms were shown to regulate the cell cycle, and many other processes as well, including stickiness, chemotaxis, phototaxis, photosynthesis, and more. In 2004, Cooper (*345*) wrote a strong rebuke of synchronized cultures, aimed mainly at methods using chemical treatments to arrest mammalian or yeast cells at certain stages of the cell cycle, advocating instead for elutriation. He proceeds to declare that whole cultures of eukaryotic cells cannot be synchronized, but totally neglects the enormous body of research supported by synchronous cultures of algae, a highly standardized model. Nonetheless, he raises good points about the potential for chemical treatments to yield results that are not physiologically relevant. However, whole culture synchronization of unicellular green algae by circadian entrainment of diel cycles is not only possible, it is an established and potent tool for studying fundamental biological processes in the algae.

Within the last few years, research on the circadian clock and cell cycle in algae has continued to make important strides forward. In 2012, Farre (*346*) thoroughly reviewed the developments of the early 21st century, which included observations of circadian regulation in unicellular and multicellular algae, as well as land plants. During this time, *A. thaliana* emerged as the dominant model for studying circadian regulation; thus the circadian clock in *A. thaliana* is very well characterized. Still, research on algal models continued, with two studies that year

focused on circadian rhythms in a coral-symbiotic alga, investigating the effects of temperature (*347*) and light (*348*). In 2014, Miyagishima *et al* (*349*) demonstrated a translation-independent mechanism of circadian control over the cell cycle in a red alga. They found that time-dependent phosphorylation of the RBR-E2F-DP complex promotes initiation of the S phase. Phosphorylation was inhibited in the light, preventing cell cycle progression. In the dark, phosphorylation of the complex is deregulated, enabling cell division. In 2015, Diamond *et al* (*350*) reported that the circadian clock in the cyanobacteria *S. elongatus* modulates metabolic flux between the Calvin cycle and the oxidative pentose phosphate pathway, using untargeted metabolomics. That same year, Noordally and Millar (*351*) reviewed advances in the understanding of circadian clocks in algae. In particular, they focused on the lens of genomic, transcriptomic, proteomic, and metabolomic (i.e. "omics") approaches. They discussed the experimental tools and mathematical modeling methods available to collect and analyze data. Understanding of the cell cycle in green algae has also grown substantially, as reviewed by Cross and Umen (*352*). However, they highlight a number of important open questions in the field, illustrating that there is still a great deal more to learn about the cell cycle and its regulators.

The development of RNA sequencing technology enabled a transcriptomics revolution that, coupled with the genomics revolution, transformed the ability to collect genome-wide datasets of gene expression. In 2014, Panchy *et al* (*353*) reported functionally distinct modules of coexpressed genes associated with diel cycles in *C. reinhardtii*. They found that about 50% of all genes in the alga undergo cyclic transcription, with a clear progression of biological processes throughout the day. The following year, another transcriptomic profile of diel cycles in *C. reinhardtii* was reported, with over 80% of all genes exhibiting cyclic transcription (*354*). They also analyzed the expression profiles of different biological processes and metabolic pathways,

finding specific signatures. The diel transcriptomic approach has also been applied to other types of photosynthetic algae, such as *N. oceanica*, a stramenopile, derived from secondary endosymbiosis of an ancestral red alga, reported by Poliner *et al* (*355*). They found that more than 60% of the genes in *N. oceanica* exhibited cyclic expression, with corresponding oscillations in lipid content. Most recently, in 2017 de los Reyes *et al* (*356*) integrated diel gene expression data from microarray and RNA-seq experiments in plants and algae to examine the evolution of diel cycles across the green lineage. Looking at cyclic transcription, they found that in the simplest species, *O. tauri*, that 90% of genes cycled, whereas in the higher plant *A. thaliana*, only 40% of genes cycled. All of these experiments clearly underscore the importance of this data for elucidating fundamental biological processes and evolutionary differentiation.

4.1.2 Conception and Purpose of Experiment

In order to capture changes in *B. braunii* gene expression and metabolite profile over time in association with light/dark (i.e. diel) cycles, a robust biomass growth experiment was developed (Figure 54). Biomass was collected every six hours (i.e. four times per day), at 5:00, 11:00, 17:00, and 23:00 each day, over the course of three days (i.e. twelve total time points). The light and dark periods were set to twelve hours each, with the lights turning on at 6:00 and off at 18:00 every day. The culture system had space to hold 36 flasks for growing algae. Each sample was collected from a single flask, and at each of the twelve time points in the experiment, three samples were collected (i.e. 36 total samples). Thus, for each time of day, there were a total of nine biological replicates collected over the three-day experiment. The biomass collected from each flask was divided into two aliquots, one for transcriptomics and one for metabolomics.

The overall experiment was divided into three phases (Figure 55). This was intended to help define key milestones, mitigate risks, provide go/no-go decision points, and ensure experiment quality. Phase 1 was a pilot study of the growth characteristics for *B. braunii* in the culture system to be used for collecting experimental samples. This phase involved testing methods for measuring culture density, determining the optimal inoculation density, and the rate of biomass growth. Phase 2 was focused on growing and collecting the experimental samples of biomass. Phase 3 was focused on preparing the experimental samples for transcriptomics and metabolomics. The remainder of this section (i.e. 4.1) will present and discuss the results of Phases 1 and 2, while the following sections (i.e. 4.2 and 4.3) will present and discuss the results of the Phase 3 transcriptomics and metabolomics analyses.

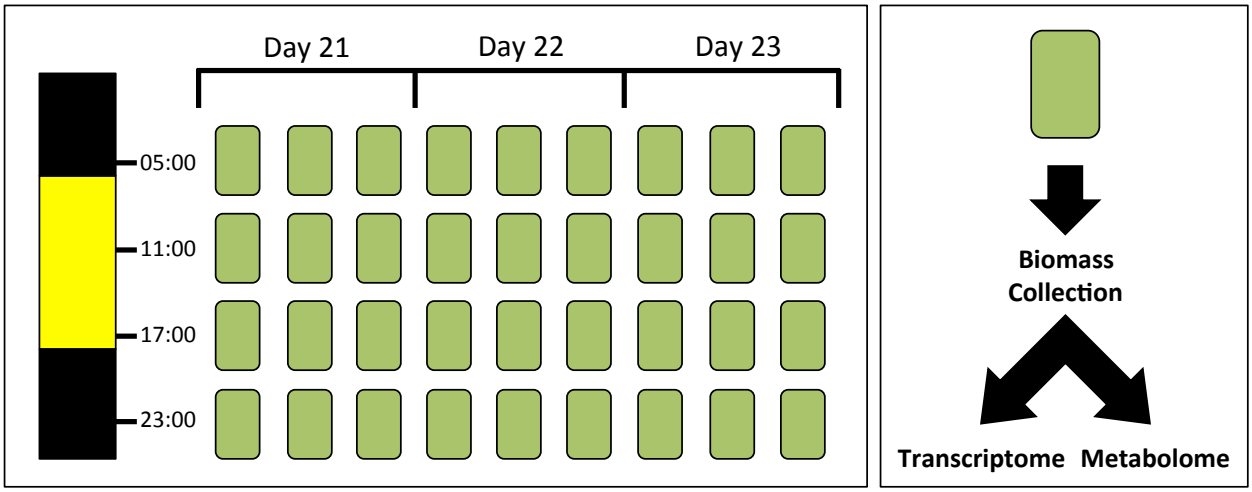


Figure 71. Overview of experimental design. This experiment was designed to capture fluctuations in transcription and metabolism in association with time. Furthermore, it was designed to determine the effects of light and dark conditions. The sample preparation strategy enables correlative analyses between the transcriptome and metabolome. This could lead to a better understanding of the impact that transcription has on metabolism. Ideally, we could also add layers of genome and proteome sequencing.

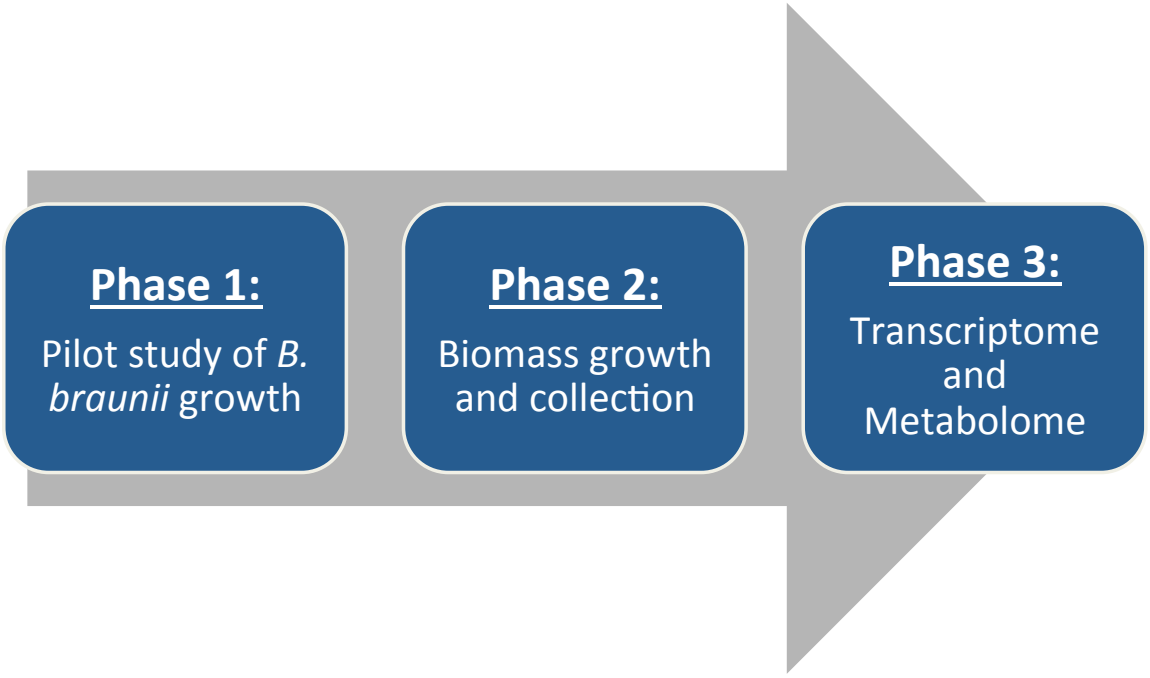


Figure 72. Division of experiment into three phases. The workflow of the experiment is designed to mitigate risk and provide a strong empirical foundation for the experimental conditions. The pilot study is important for determining optimal parameters for the experiment. The experimental conditions (i.e. number of replicates per condition, number of collections, time of collections, etc) will determine the value of the data.

4.2 Materials and Methods

The Materials and Methods for this section are described in Appendix C.

4.3 Results and Discussion

This section begins by reviewing the experimental setup including the pilot phase and experimental growth phase. The gene expression data and metabolomics data are then presented and discussed in detail.

4.3.1 Experimental Design and Biomass Collection

This section describes the experimental planning process in the initial determination of conditions and setup and collection of the experimental samples.

4.3.1.1 Pilot Testing the Culture System

One of the most essential methods in this experiment is the measurement of biomass density in the algal cultures (Figure 56). The first step is taking an aliquot of the culture and collecting the biomass onto a pre-weighed filter by vacuum filtration. It is important to select an appropriate filter type to ensure that algal biomass does not pass through the filter. The next step is drying the biomass in an oven, and then weighing the filter with the dried biomass This enables the determination of the dry mass that was present in the aliquot and in turn the extrapolation of culture density. However, algal cultures are not completely homogenous and thus an aliquot may not have density representative of the whole culture, especially if mixing is poor. One important question with regard to aliquots is: what volume of aliquot should be taken? This is partially dependent on the total amount of solution available, as an aliquot should not remove a substantial

portion of the culture. To determine the impact of aliquot volume on density measurements, several aliquots of 50 mL (n = 3) and 10 mL (n = 3) were taken from a single flask of algal culture (Figure 57). This experiment revealed that 50 mL aliquots give more accurate measurements of culture density. Thus measurements of culture density in this experiment should be taken with 50 mL aliquots.

The inoculation density is one of the key variables in the experimental equation. In order to determine the effect of inoculation density on growth rate, four different inoculation densities were tested (Table 31). The inoculant culture had a density of 2.3 g/L and was diluted as required. For each density condition, three flasks of culture were inoculated to the target density. Aliquots of the cultures were taken at days 3, 8, 12, 19, 26, and 42 after inoculation, and the density of the cultures was measured (Figure 58). This experiment revealed that the growth rate was very slow, but not significantly affected by the inoculation density. The most important lesson learned from this experiment was that cultures in the center of the rack had a higher growth rate. This was due to the fact that incident light from neighboring light bulbs was more intense in the center of the rack than on the edges of the rack. This observation led to the conclusion that light barriers must be placed between the flasks to ensure relatively uniform lighting conditions.

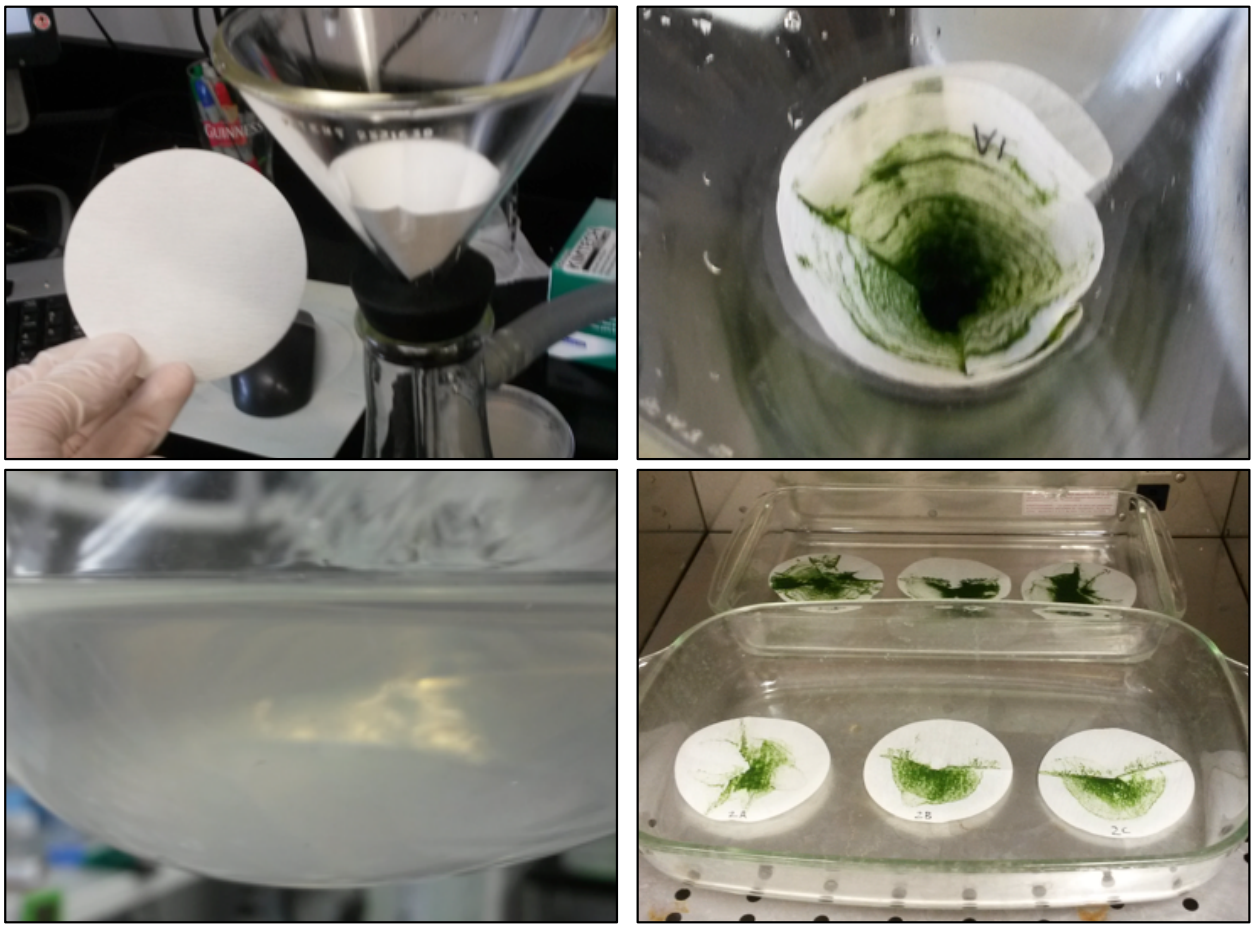


Figure 73. Method used to determine density of cultures. These images show the filtration method utilized to collect samples of algae from the media. A pre-weighed paper filter is placed in a conical funnel that is attached to a vacuum flask. The vacuum is drawn and a sample of culture is applied to the filter. The filters are then placed in trays and dried at 85 °C for 2 hours. The filters are allowed to cool and the samples are weighed.

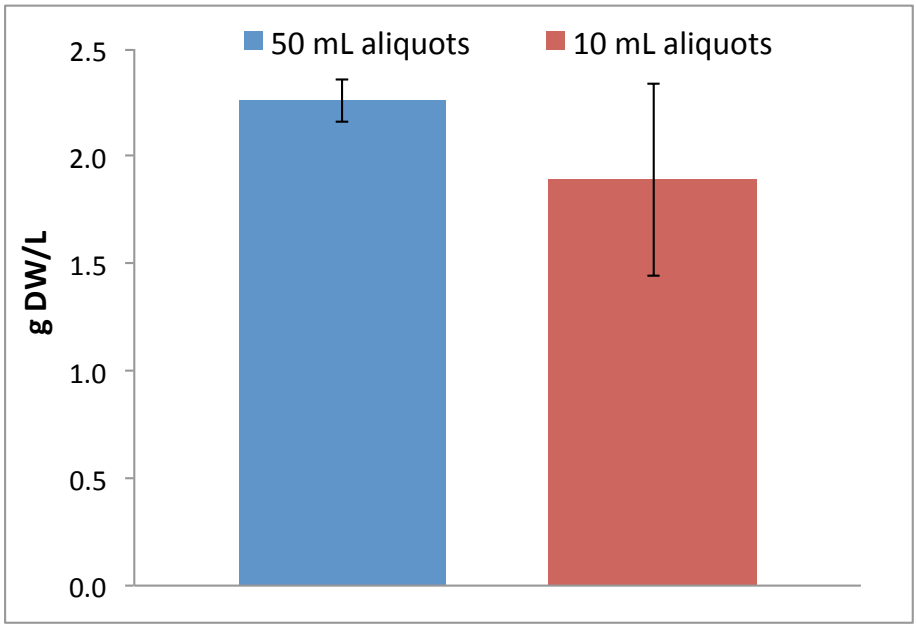


Figure 74. Impact of aliquot size on density measurements. These data show that larger sample sizes result in a smaller amount of error. This is important for obtaining accurate measurements of culture density.

Table 29. Summary of pilot testing inoculation scheme. The pilot experiment was primarily designed to learn information about the effect of inoculation density on the growth rate. Four conditions were devised to test a range of low to high density (0.05–0.40 g/L).

Sample Name	Inoculation Density (g/L)	Culture Volume (L)	Biomass Required (g)	Inoculant Volume (mL)	Water Volume (mL)
LL	0.05	0.95	0.0475	21	179
LH	0.10	0.95	0.0950	42	158
HL	0.20	0.95	0.1900	84	116
HH	0.40	0.95	0.3800	168	32

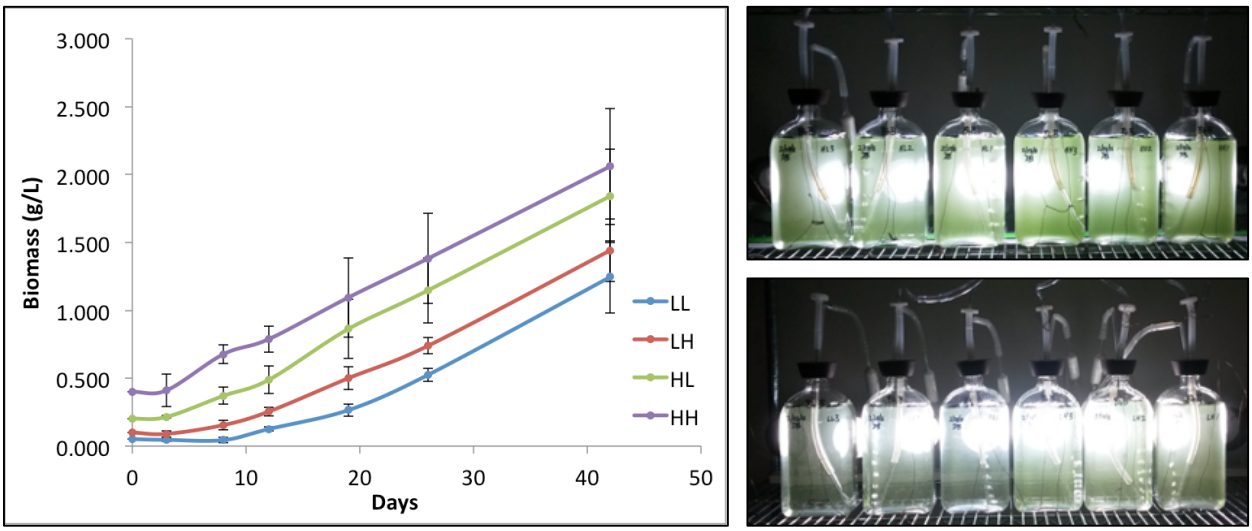


Figure 75. Summary of growth curve results from pilot scale testing. The maximum density achieved in the experiment was approximately 2.25 g/L, resulting from the highest inoculation density (0.40 g/L) after 42 days. The flask location had a notable impact on the reults, with flasks in the center of the rack growing to higher densities. This is likely because the center area of the rack had the highest incidence of light. Because of this, separators were placed between all the flasks in the experimental growth phase.

4.3.1.2 Culturing Biomass for Experiment

Based on the pilot results, a target inoculation density of 0.2 g/L was selected for the experiment. Four flasks of culture were grown for a month and then mixed together to prepare the inoculant for the 36 experimental cultures (Figure 59). The density of the inoculant was 1.9 g/L and each experimental flask had 861 mL of media. To each flask, 89 mL of inoculant was added, bringing the final volume to 950 mL of culture. The flasks were then loaded onto the cultivation racks, connected to the air supply system (2.5% CO_2 supplement), which also provides mixing to the cultures via bubbling. The cultures were allowed to grow for three weeks, at which point began the collection of experimental samples. Biomass was harvested by vacuum filtration with a 10 μm mesh nylon filter. The biomass was scraped off the filter with a stainless-steel spatula and split between two tubes (Table 32). The tubes were weighed and immediately frozen in liquid nitrogen and the frozen biomass was then stored at -80 °C until further use. On average, 898 mg of biomass was collected in total. Immediately prior to harvesting the biomass, a 50 mL aliquot of each culture was taken in order to determine the biomass density (Figure 60). Given that the starting density was 0.2 g/L, by measuring the density at harvest, the doubling time of the biomass could be calculated. On average, the culture density at harvest was 0.7 g/L and the doubling time for the cultures was 13 days. Looking at the individual culture densities, there does not appear to be a trend of biomass increasing with harvest time. Similarly, calculating averages for each time point did not appear to reveal any trends (Figure 61). However, calculating average culture density for each day did show a slight trend of biomass increasing with day, although there were large standard deviations. Ultimately, the results are fairly consistent and provide a reasonable amount of biomass for experimentation.

Figure 76. Setup for collection of experimental samples of biomass. A total 36 flasks of media were inoculated to a uniform density with a homogenous inoculant. The inoculant was derived from four high-density flasks of *B. braunii* race B (Showa). The flasks were adjusted to ensure an approximately even amount of light and cardboard separators were placed between all the flasks.

Table 30. Summary of biomass collection for experimentation. This table shows the amounts of biomass collected in each sample, as well as the culture density at the time of harvest.

Day	Time	Sample	Dry Mass (mg)	Culture Density (g/L)	Tube A (mg)	Tube B (mg)	Total Biomass (mg)
21	5:00 AM	1	35.6	0.712	330	743	1073
		2	34.9	0.698	384	577	961
		3	36.5	0.730	377	612	989
	11:00 AM	4	32.7	0.654	255	392	647
		5	28.2	0.564	342	250	592
		6	27.5	0.550	338	519	857
	5:00 PM	7	26.7	0.534	245	458	703
		8	26.0	0.520	256	484	740
		9	31.9	0.638	237	406	643
	11:00 PM	10	30.5	0.610	254	650	904
		11	31.7	0.634	244	754	998
		12	27.3	0.546	267	571	838
22	5:00 AM	13	29.5	0.590	343	431	774
		14	34.0	0.680	329	542	871
		15	33.9	0.678	378	413	791
	11:00 AM	16	30.3	0.606	376	532	908
		17	35.4	0.708	352	595	947
		18	33.2	0.664	331	577	908
	5:00 PM	19	31.8	0.636	280	779	1059
		20	30.7	0.614	243	501	744
		21	32.5	0.650	248	790	1038
	11:00 PM	22	37.2	0.744	249	776	1025
		23	34.6	0.692	267	616	883
		24	31.3	0.626	277	658	935
23	5:00 AM	25	39.7	0.794	367	697	1064
		26	42.7	0.854	379	791	1170
		27	37.0	0.740	381	651	1032
	11:00 AM	28	40.9	0.818	276	914	1190
		29	34.1	0.682	250	556	806
		30	31.2	0.624	273	483	756
	5:00 PM	31	35.4	0.708	264	671	935
		32	27.9	0.558	265	367	632
		33	29.4	0.588	253	433	686
	11:00 PM	34	28.9	0.578	252	574	826
		35	39.3	0.786	250	952	1202
		36	45.9	0.918	272	933	1205

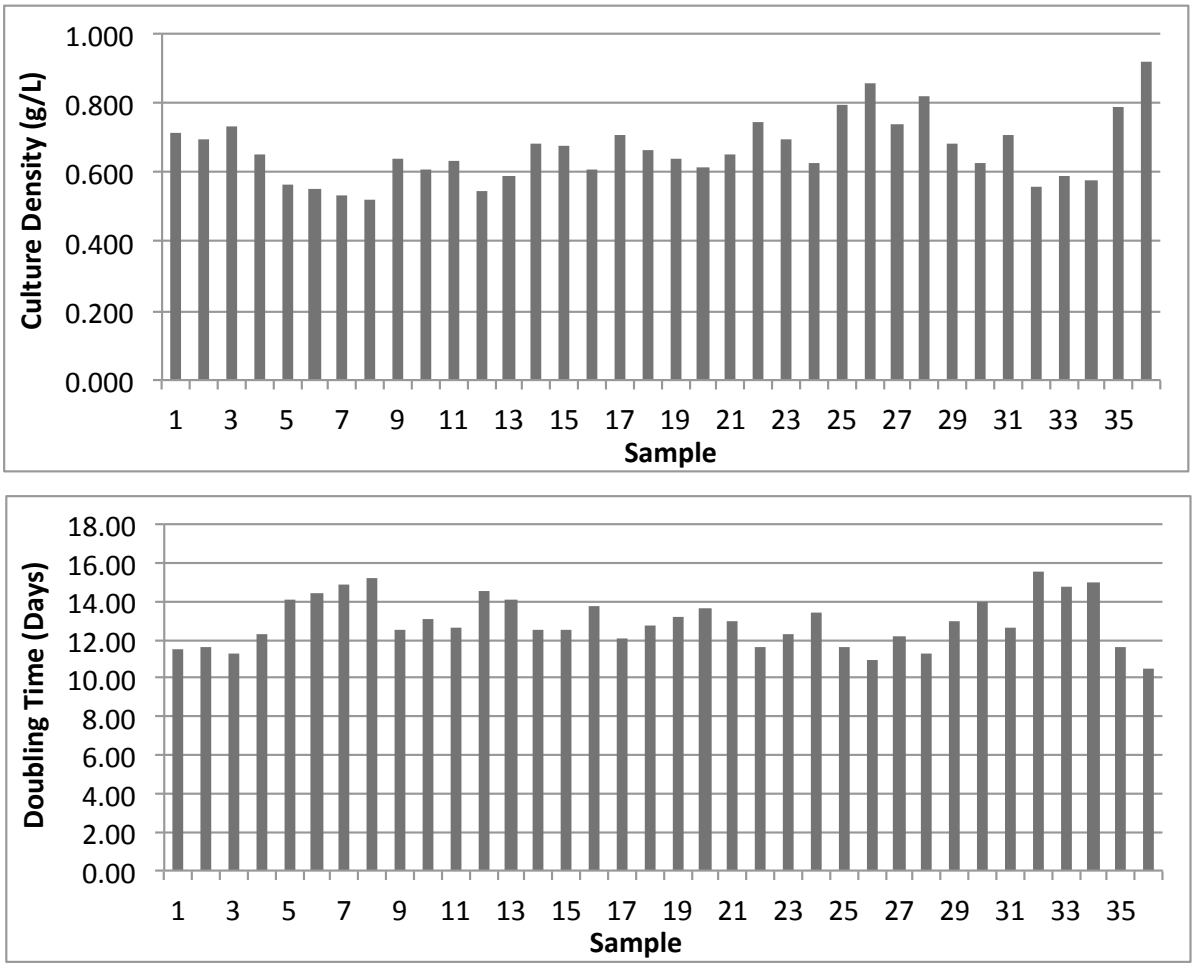


Figure 77. Summary of culture density and doubling time for experimental samples. These data show the variation in culture density and the associated doubling time throughout the experiment. These differences could have a substantial impact on the results of the experiment.

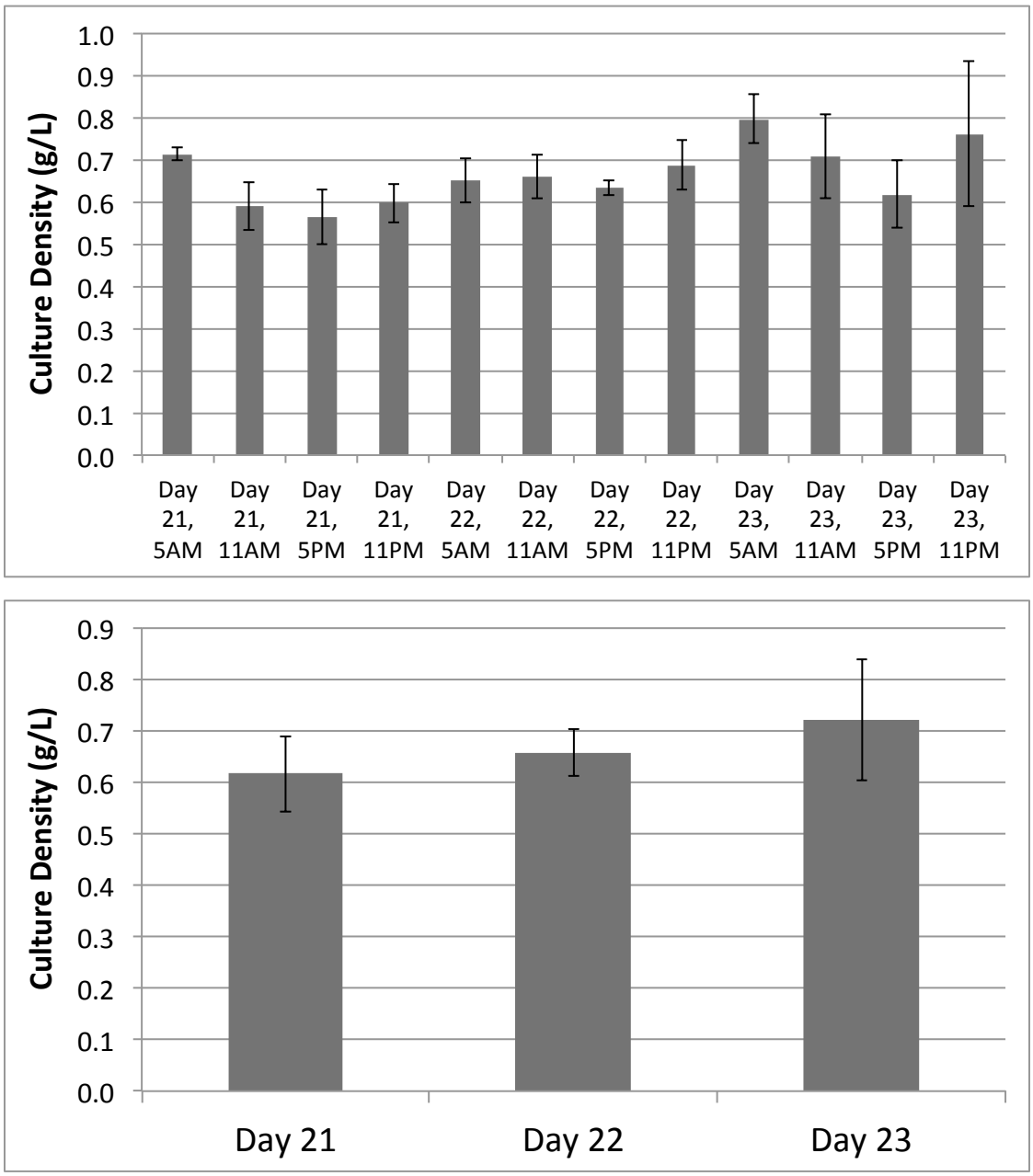


Figure 78. Analysis of culture density by time point and by day. By averaging the culture density of the samples according to their time of day or day of harvest, the trend of growth over time becomes clearer. These data suggest that the algae are slowly and continuously growing, rather than growing in bursts.

4.3.2 Analysis of Gene Expression

The following sections describe the processes of RNA extraction and sequencing, differential gene expression analysis, and determination of gene coexpression clusters. These are the most fundamental aspects of a systematic analysis of the transcriptome. Finally, the key pathways that were discussed in section 3 are analyzed for patterns of gene expression. These data provide an important window into the functional interpretation of genomic information.

4.3.2.1 RNA Extraction and Sequencing Results

With the biomass collected, the next phase of the experiment was RNA extraction and sample preparation. Samples were prepared in a stepwise process to minimize differences in processing between samples. First, all of the samples were ground into a fine powder with a mortar and pestle under liquid nitrogen. The complete sample of biomass in each tube was pulverized; and 60-90 mg aliquots of the frozen, powdered biomass were immediately added to pre-weighed tubes of TRIzol reagent (ThermoFisher). After adding biomass to each tube, the samples were incubated at 42 °C for 15 minutes to promote dissolution of the biomass into the TRIzol. Each sample was then spun down at 16,000 x g for 2 minutes at room temperature and the supernatants were transferred to fresh, respectively labeled tubes with glass Pasteur pipettes. The TRIzol samples were then stored at -80 °C until further processing. The remaining frozen biomass powder was carefully transferred back into a frozen tube and stored at -80 °C for future use. For each sample, two tubes of biomass dissolved in TRIzol were prepared, in order to have a backup in case of primary sample failure or insufficient yield (Table 33). In the first set of TRIzol samples, there was an average of 79.4±13.9 mg of biomass; while in the second set there was an average of 61.0±2.3 mg of biomass.

Once all of the biomass had been pulverized and stored in TRIzol, the RNA extraction process was initiated. Each sample was removed from the -80 °C freezer and thawed at 37 °C with occasional vortexing. Then 200 µL of chloroform was added to each sample, followed by thorough vortexing. The samples were incubated for 2 minutes at room temperature and then spun down at 16,000 x *g* for 5 minutes at room temperature. About 400 µL of the upper aqueous phase was carefully transferred to fresh, respectively labeled tubes. Then 1 mL of 100% isopropanol was added to each sample, followed by gentle mixing by inversion. The samples were spun down at 16,000 x *g* for 10 minutes at room temperature, and the supernatants were carefully decanted into a waste beaker. To separate polysaccharides from the RNA, 1 mL of 2 M LiCl was added to each sample, followed by thorough pipetting and vortexing, until the pellets were highly fragmented. The samples were then spun down at 16,000 x *g* for 10 minutes at room temperature, and the supernatants were removed by pipette. The lithium chloride wash step was repeated two more times. Then 1 mL of 75% ethanol (25% distilled, deionized, and autoclaved water) was added to each sample and the pellets were broken apart by pipette, followed by brief vortexing. The samples were then spun down at 16,000 x *g* for 5 minutes at room temperature, and the supernatants were carefully decanted into a waste beaker. The samples were then dried under a gentle stream of air, and the pellets were resuspended in 100 µL of pure dH_2O. Finally, 300 µL of TRIzol and 400 µL of 100% ethanol were added to each sample, followed by re-purification with the Direct-zol RNA MiniPrep Plus kit (Zymo Research). Importantly, this kit included a DNase digestion step to remove contaminating genomic DNA from the total RNA samples. The quantity and quality of purified RNA was assessed by absorbance spectrophotometry using an Epoch microplate reader (BioTek Instruments). The average amount of RNA extracted was 6.3±2.2 µg, providing more than enough sample for quality analysis and library preparation by the JGI (Table 34). The $A_{260/280}$

and $A_{260/230}$ ratios of the samples indicated that there was relatively little contamination by proteins or carbohydrates, respectively.

The total RNA samples were subsequently diluted to uniform concentrations of 30 ng/μL in 100 μL, for a total of 3 μg of RNA (Table 34). The dilutions were then loaded onto a pre-labeled 96-well plate from the JGI, and shipped back to the JGI overnight on dry ice. Quality analysis of the RNA samples was performed by the JGI using a BioAnalyzer and Qubit, and all samples passed the required quality threshold. Sequencing libraries were prepared by the JGI, first enriching for mRNAs with a polyA-tail selection. The mRNA-enriched samples were then processed with the standard Illumina TruSeq strand-specific, high sample library preparation protocol. The libraries were multiplexed into groups of twelve and then sequenced on three separate lanes with an Illumina HiSeq-2500 1TB sequencer with 2x150 bp chemistry (Table 35). The result was 1.1 billion read pairs (i.e. 2.2 billion total reads), with an average of 66.2 million reads per library. The sequencing depth is reasonably consistent across all the samples, with the exception of sample 20 (day 22, 11:00 AM), which had about three times the average number of reads. In total, after removing low quality reads and trimming low quality bases from the 3'-ends of the reads, there were 329.4 billion bases of sequence data. The scale of this data is unparalleled in the field of *B. braunii* research and represents a massive step forward in both quantity and quality of sequencing data for *B. braunii*.

Table 31. Preparation of TRIzol solutions with ground biomass. These data show the amount of biomass used to prepare each RNA sample. Despite best efforts to prepare uniform amounts of biomass, there is some variation.

Sample	TZ1 Empty (mg)	TZ1 Full (mg)	TZ1 Biomass (mg)	TZ2 Empty (mg)	TZ2 Full (mg)	TZ2 Biomass (mg)
1	2054.0	2144.8	90.8	2050.6	2110.7	60.1
2	2045.4	2135.8	90.4	2038.2	2101.1	62.9
3	2036.8	2117.8	81.0	2040.4	2100.0	59.6
4	2046.1	2125.7	79.6	2042.0	2102.7	60.7
5	2042.5	2133.2	90.7	2044.8	2103.6	58.8
6	2049.1	2153.2	104.1	2056.7	2116.7	60.0
7	2035.6	2112.2	76.6	2059.0	2119.1	60.1
8	2045.4	2145.1	99.7	2044.1	2106.1	62.0
9	2047.4	2129.8	82.4	2066.1	2130.2	64.1
10	2054.6	2151.4	96.8	2052.9	2112.5	59.6
11	2040.3	2153.6	113.3	2033.5	2094.6	61.1
12	2039.7	2130.8	91.1	2048.6	2120.8	72.2
13	2055.3	2121.4	66.1	2062.7	2124.0	61.3
14	2040.0	2105.4	65.4	2050.7	2110.8	60.1
15	2049.3	2118.5	69.2	2060.2	2120.5	60.3
16	2049.9	2116.2	66.3	2049.6	2112.3	62.7
17	2042.6	2120.0	77.4	2063.3	2123.1	59.8
18	2045.5	2116.5	71.0	2051.4	2110.9	59.5
19	2041.6	2126.8	85.2	2048.4	2110.4	62.0
20	2039.4	2104.2	64.8	2049.1	2108.6	59.5
21	2046.9	2110.4	63.5	2043.7	2105.6	61.9
22	2045.8	2120.9	75.1	2038.5	2098.0	59.5
23	2045.5	2112.1	66.6	2048.5	2107.5	59.0
24	2038.1	2104.9	66.8	2034.7	2095.2	60.5
25	2039.6	2143.5	103.9	2062.5	2124.5	62.0
26	2042.5	2121.9	79.4	2056.2	2115.5	59.3
27	2036.2	2103.1	66.9	2057.7	2118.0	60.3
28	2044.6	2115.9	71.3	2053.1	2112.4	59.3
29	2048.7	2119.2	70.5	2054.8	2115.4	60.6
30	2042.5	2116.9	74.4	2055.6	2117.2	61.6
31	2047.3	2117.6	70.3	2072.8	2134.4	61.6
32	2044.5	2116.6	72.1	2050.7	2112.3	61.6
33	2051.4	2135.4	84.0	2048.9	2108.9	60.0
34	2044.1	2118.3	74.2	2065.4	2126.7	61.3
35	2041.8	2119.3	77.5	2072.2	2132.4	60.2
36	2052.6	2133.9	81.1	2062.3	2124.7	62.4

Table 32. Extracting RNA and preparing samples for JGI. This table shows the dilutions that were prepared from the total RNA samples. The samples were diluted to a uniform concentration of 30 ng/μL in 100 μL for a total 3 μg of RNA per sample.

Sample	260/280	260/230	ng/uL	uL	ng	uL RNA	uL dH2O	Well ID
1	2.10	2.19	67.82	98	6646	44.2	55.8	B1
2	2.13	2.58	88.46	98	8669	33.9	66.1	C1
3	2.08	2.03	60.36	98	5915	49.7	50.3	D1
4	2.06	2.20	41.26	98	4043	72.7	27.3	E1
5	2.09	2.23	113.93	98	11166	26.3	73.7	F1
6	2.08	2.31	136.25	98	13352	22.0	78.0	G1
7	2.07	2.43	72.52	98	7107	41.4	58.6	A2
8	2.07	2.33	76.70	98	7517	39.1	60.9	B2
9	2.06	2.24	76.53	98	7500	39.2	60.8	C2
10	2.06	2.19	40.28	98	3948	74.5	25.5	D2
11	2.08	2.43	57.38	98	5623	52.3	47.7	E2
12	2.08	2.70	70.47	98	6906	42.6	57.4	F2
13	2.06	2.15	36.51	98	3578	82.2	17.8	G2
14	2.09	2.34	36.44	98	3571	82.3	17.7	H2
15	2.09	2.32	39.47	98	3868	76.0	24.0	A3
16	2.06	2.13	48.72	98	4775	61.6	38.4	B3
17	2.09	2.25	56.22	98	5510	53.4	46.6	C3
18	2.06	2.21	85.76	98	8404	35.0	65.0	D3
19	1.94	1.84	31.49	98	3086	95.3	4.7	E3
20	1.96	2.24	47.73	98	4678	62.9	37.1	F3
21	1.95	2.09	41.73	98	4090	71.9	28.1	G3
22	2.04	2.17	50.81	98	4979	59.0	41.0	H3
23	2.06	2.25	60.19	98	5899	49.8	50.2	A4
24	2.06	2.36	82.75	98	8110	36.3	63.7	B4
25	2.08	2.09	51.08	98	5006	58.7	41.3	C4
26	2.08	2.25	64.30	98	6301	46.7	53.3	D4
27	2.09	2.31	43.90	98	4302	68.3	31.7	E4
28	2.07	2.25	104.82	98	10272	28.6	71.4	F4
29	1.98	2.23	82.70	98	8105	36.3	63.7	G4
30	2.07	2.33	66.34	98	6501	45.2	54.8	H4
31	2.07	2.11	56.85	98	5571	52.8	47.2	A5
32	2.06	2.20	49.21	98	4823	61.0	39.0	B5
33	2.08	2.34	65.54	98	6423	45.8	54.2	C5
34	2.08	2.27	75.11	98	7361	39.9	60.1	D5
35	2.06	2.13	48.98	98	4800	61.2	38.8	E5
36	2.08	2.36	74.28	98	7279	40.4	59.6	F5

Table 33. Summary of RNA-seq libraries from Illumina HiSeq-2500 1TB. Only two of the samples (8 and 10) failed in the library preparation phase. The remaining 34 sample yielded fairly consistent sequencing results, except for sample 20.

Sample	Name	Input Reads	Remaining Reads	Percent Remaining	Input Bases	Remaining Bases	Percent Remaining
1	D21_05AM_R1	50,610,416	50,238,766	99.27%	1.50E+10	7.31E+09	48.57%
2	D21_05AM_R2	42,757,580	42,493,244	99.38%	1.27E+10	6.19E+09	48.61%
3	D21_05AM_R3	71,411,186	70,904,154	99.29%	2.12E+10	1.03E+10	48.53%
4	D21_11AM_R1	63,424,378	63,061,836	99.43%	1.88E+10	9.13E+09	48.53%
5	D21_11AM_R2	67,976,688	67,169,630	98.81%	2.02E+10	9.76E+09	48.28%
6	D21_11AM_R3	67,821,870	67,237,230	99.14%	2.01E+10	9.74E+09	48.42%
7	D21_05PM_R1	52,900,188	52,525,394	99.29%	1.58E+10	7.72E+09	48.81%
8	D21_05PM_R2	NA	NA	NA	NA	NA	NA
9	D21_05PM_R3	43,751,548	43,423,794	99.25%	1.31E+10	6.35E+09	48.51%
10	D21_11PM_R1	57,582,364	57,106,258	99.17%	1.72E+10	8.33E+09	48.44%
11	D21_11PM_R2	NA	NA	NA	NA	NA	NA
12	D21_11PM_R3	57,274,130	56,688,358	98.98%	1.71E+10	8.26E+09	48.33%
13	D22_05AM_R1	64,833,008	64,505,318	99.49%	1.94E+10	9.48E+09	48.87%
14	D22_05AM_R2	85,494,374	84,818,506	99.21%	2.55E+10	1.24E+10	48.62%
15	D22_05AM_R3	54,064,326	53,592,206	99.13%	1.62E+10	7.88E+09	48.76%
16	D22_11AM_R1	73,869,432	73,344,114	99.29%	2.21E+10	1.07E+10	48.68%
17	D22_11AM_R2	48,357,574	48,037,050	99.34%	1.45E+10	7.08E+09	48.91%
18	D22_11AM_R3	72,021,744	71,640,046	99.47%	2.15E+10	1.05E+10	48.86%
19	D22_05PM_R1	53,146,662	52,775,032	99.30%	1.59E+10	7.77E+09	48.87%
20	D22_05PM_R2	178,080,508	176,988,754	99.39%	5.32E+10	2.59E+10	48.60%
21	D22_05PM_R3	66,810,236	66,372,102	99.34%	2.00E+10	9.76E+09	48.86%
22	D22_11PM_R1	56,449,216	56,097,168	99.38%	1.69E+10	8.19E+09	48.58%
23	D22_11PM_R2	62,923,560	62,527,254	99.37%	1.88E+10	9.18E+09	48.82%
24	D22_11PM_R3	55,644,236	55,098,016	99.02%	1.66E+10	8.05E+09	48.45%
25	D23_05AM_R1	64,898,398	64,585,206	99.52%	1.94E+10	9.45E+09	48.81%
26	D23_05AM_R2	82,919,056	82,332,744	99.29%	2.48E+10	1.21E+10	48.81%
27	D23_05AM_R3	49,835,152	49,551,332	99.43%	1.49E+10	7.26E+09	48.75%
28	D23_11AM_R1	64,248,820	63,779,204	99.27%	1.92E+10	9.34E+09	48.69%
29	D23_11AM_R2	65,325,952	64,902,832	99.35%	1.95E+10	9.50E+09	48.71%
30	D23_11AM_R3	76,238,240	75,720,056	99.32%	2.28E+10	1.11E+10	48.70%
31	D23_05PM_R1	66,169,218	65,472,206	98.95%	1.98E+10	9.65E+09	48.69%
32	D23_05PM_R2	81,412,526	80,915,946	99.39%	2.44E+10	1.19E+10	48.84%
33	D23_05PM_R3	52,903,254	52,470,780	99.18%	1.58E+10	7.72E+09	48.76%
34	D23_11PM_R1	74,669,484	74,243,940	99.43%	2.24E+10	1.09E+10	48.89%
35	D23_11PM_R2	58,979,084	58,473,654	99.14%	1.77E+10	8.61E+09	48.75%
36	D23_11PM_R3	80,688,270	80,300,338	99.52%	2.42E+10	1.18E+10	48.95%

4.3.2.2 Quality Control of Biological Replicates

Alignments against the genome and transcriptome with HISAT2 were used to assess library quality (Table 36). First, an index of the *B. braunii* version 2.0 genome was created with hisat2-build, supplied with exon coordinates and splice sites from the v2.1 annotations. Second, another index was created for the genome without the exon coordinates and splice sites built in. Third, an index was created for the predicted transcriptome from the v2.1 annotations, with no exon or splice information. The HISAT2 alignment "v1" included the option --tmo (i.e. map only against known transcriptome) and used the index built with exons and splice sites. The resulting average of 23% alignment was surprisingly low. When using an index built without exons and splice sites and the --tmo option is omitted (alignment "v2"), an average 93% of reads align against the genome. To check for index effects, the alignment "v3" omitted the --tmo option, but used an index built with exons and splice sites. Furthermore, the alignment "v4" included the option --tmo and used an index built without exons and splice sites. The data show that inclusion of exons and splice sites in the index essentially has no effect on the alignment when the --tmo option is omitted. Interestingly, when aligning reads against the predicted transcriptome (alignment "v5") using a standard index and including the --no-spliced-alignment option, an average 70% of reads were aligned. This indicates that inclusion of the option --tmo severely limits the alignment and may not accurately represent reads in the library. Moreover, alignment of 70% against the transcriptome and 93% against the genome suggests that there are a substantial amount of non-coding sequences included in the libraries. These data raise important questions about the viability of quantification methods reliant on read alignment.

The predicted transcriptome from the v2.1 annotations was quantified with kallisto (*357*), which is integrated into the standard Trinity RNA-seq analysis pipeline (*358*). This method has

the advantage of being alignment-free, relying instead on k-mer analyses to quantify transcripts. The number of fragments per library reflects the sequencing depth (Figure 62). Due to the poor quality of samples 4-6, indicated by lower than average alignments, and the excessively high sequencing depth of sample 20, these libraries were omitted from downstream analyses, as they could obfuscate the results. The sample correlation matrix was improved when the low-quality samples were omitted from the quantification procedure (Figure 63). The samples were strongly clustered according to the time of day that they were collected. Principal component analysis of the samples before and after library QC also showed that time of day was the major factor impacting gene expression (Figure 64). However, this also showed some substantial variation in the samples collected at the 5:00 AM time point after removal of the low-quality samples. These data demonstrate the importance of having sufficient biological replicates built into the experiment in order to account for loss of samples due to quality issues. They also clearly show that time of day is a major factor affecting the expression patterns of genes.

Table 34. Alignment of RNA-seq libraries against the genome and transcriptome. These data show that the libraries are of similar quality, except for samples 4, 5, and 6, which have much lower rates of read alignment.

Sample	Library	Alignment v1	Alignment v2	Alignment v3	Alignment v4	Alignment v5
1	D21_0500_R1	22.52%	92.73%	93.31%	0.00%	69.88%
2	D21_0500_R2	23.35%	93.66%	94.21%	0.00%	70.80%
3	D21_0500_R3	22.04%	93.15%	93.70%	0.00%	69.95%
4	D21_1100_R1	12.62%	65.22%	65.58%	0.00%	47.31%
5	D21_1100_R2	11.40%	57.35%	57.69%	0.00%	41.40%
6	D21_1100_R3	5.74%	42.50%	42.75%	0.00%	29.21%
7	D21_1700_R1	23.50%	93.76%	94.29%	0.00%	70.73%
9	D21_1700_R3	22.72%	93.37%	93.92%	0.00%	70.14%
10	D21_2300_R1	21.75%	92.85%	93.47%	0.00%	68.82%
12	D21_2300_R3	21.79%	93.22%	93.81%	0.00%	68.36%
13	D22_0500_R1	22.64%	93.98%	94.57%	0.00%	70.40%
14	D22_0500_R2	23.24%	93.67%	94.25%	0.00%	70.75%
15	D22_0500_R3	20.61%	84.44%	84.93%	0.00%	63.70%
16	D22_1100_R1	24.53%	93.54%	94.09%	0.00%	70.90%
17	D22_1100_R2	24.33%	93.51%	94.03%	0.00%	71.37%
18	D22_1100_R3	24.06%	93.74%	94.24%	0.00%	70.65%
19	D22_1700_R1	24.98%	93.80%	94.37%	0.00%	72.12%
20	D22_1700_R2	23.96%	93.57%	94.10%	0.00%	71.45%
21	D22_1700_R3	23.86%	94.43%	94.98%	0.00%	72.02%
22	D22_2300_R1	22.08%	92.48%	93.11%	0.00%	69.19%
23	D22_2300_R2	21.78%	92.48%	93.11%	0.00%	69.11%
24	D22_2300_R3	22.62%	93.26%	93.85%	0.00%	70.54%
25	D23_0500_R1	21.58%	93.89%	94.55%	0.00%	69.80%
26	D23_0500_R2	23.27%	94.50%	95.14%	0.00%	71.04%
27	D23_0500_R3	21.64%	93.98%	94.65%	0.00%	69.51%
28	D23_1100_R1	23.68%	89.61%	90.16%	0.00%	68.69%
29	D23_1100_R2	24.09%	93.62%	94.17%	0.00%	71.29%
30	D23_1100_R3	24.54%	94.25%	94.78%	0.00%	71.63%
31	D23_1700_R1	24.70%	93.54%	94.10%	0.00%	71.57%
32	D23_1700_R2	24.34%	91.64%	92.19%	0.00%	70.14%
33	D23_1700_R3	24.30%	92.12%	92.66%	0.00%	70.66%
34	D23_2300_R1	22.56%	92.65%	93.27%	0.00%	68.93%
35	D23_2300_R2	22.65%	93.83%	94.48%	0.00%	70.14%
36	D23_2300_R3	22.59%	92.71%	93.40%	0.00%	69.12%
	Average	23.11%	92.97%	93.54%	0.00%	70.11%

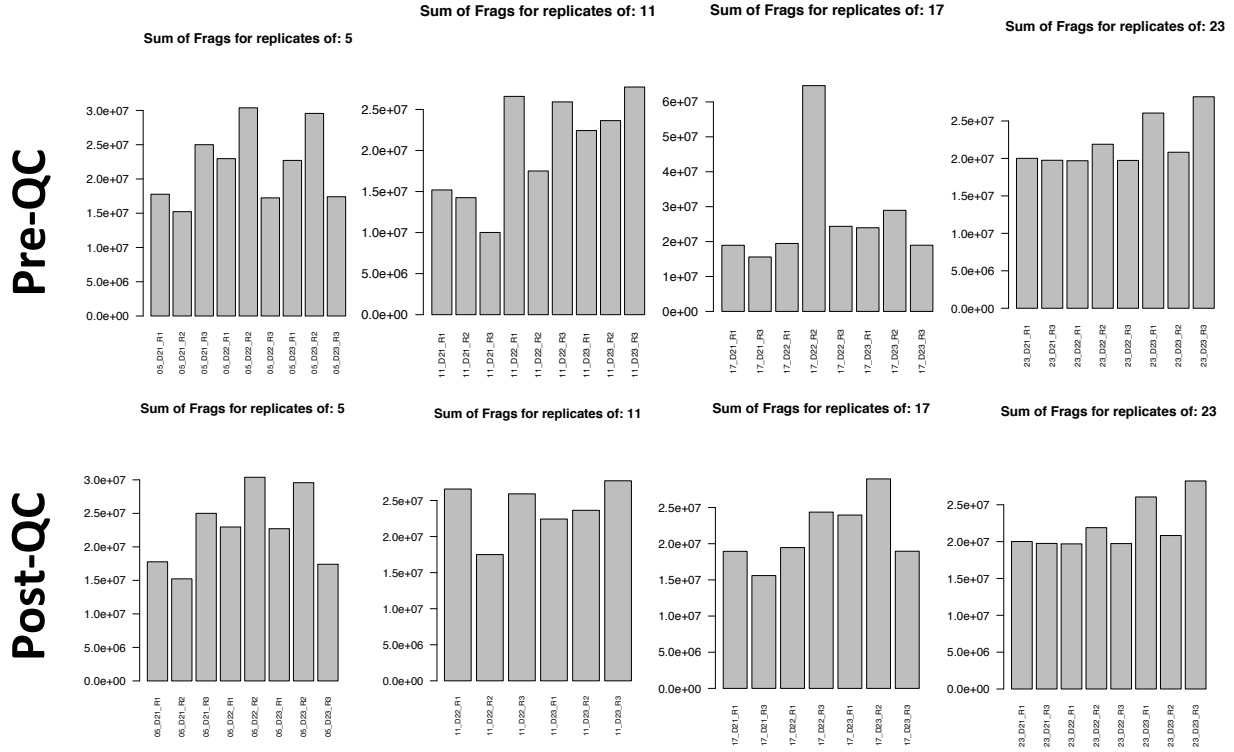


Figure 79. Quantification of *B. braunii* transcripts before and after QC of libraries. These data show that the removal of low-quality sequencing libraries results in a more even distribution of fragments across the sample replicates. The 11:00 and 17:00 time points are particularly affected by the QC filtering because they had the most low-quality libraries.

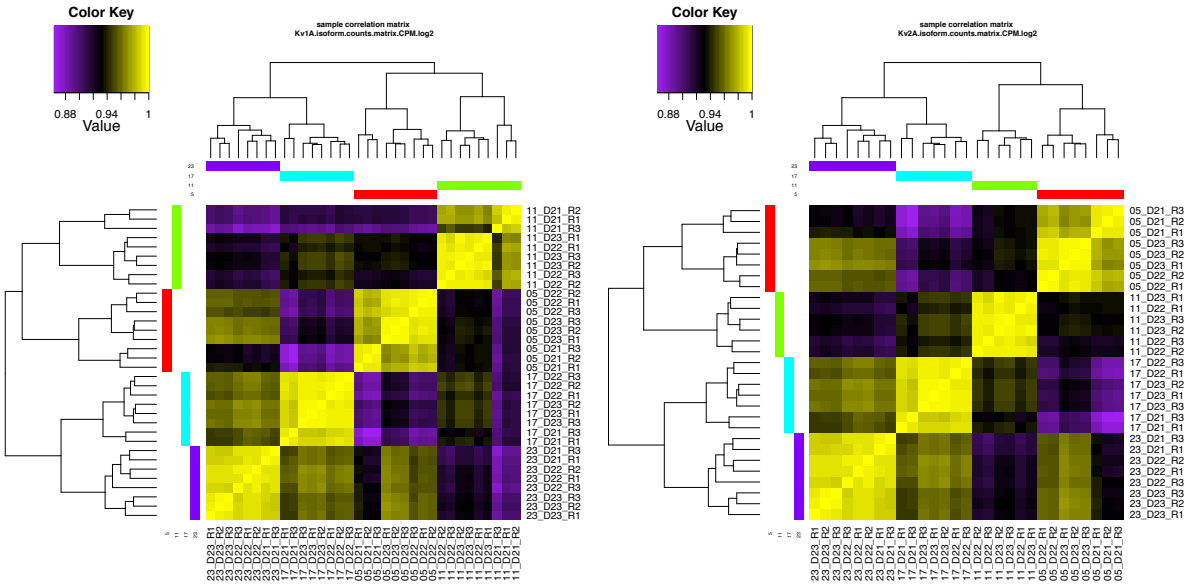


Figure 80. Sample correlation matrix before and after QC of libraries. These data show that removal of the low-quality sequencing libraries results in better sample correlation. This indicates that it is important to remove these libraries so as to obtain more consistent results.

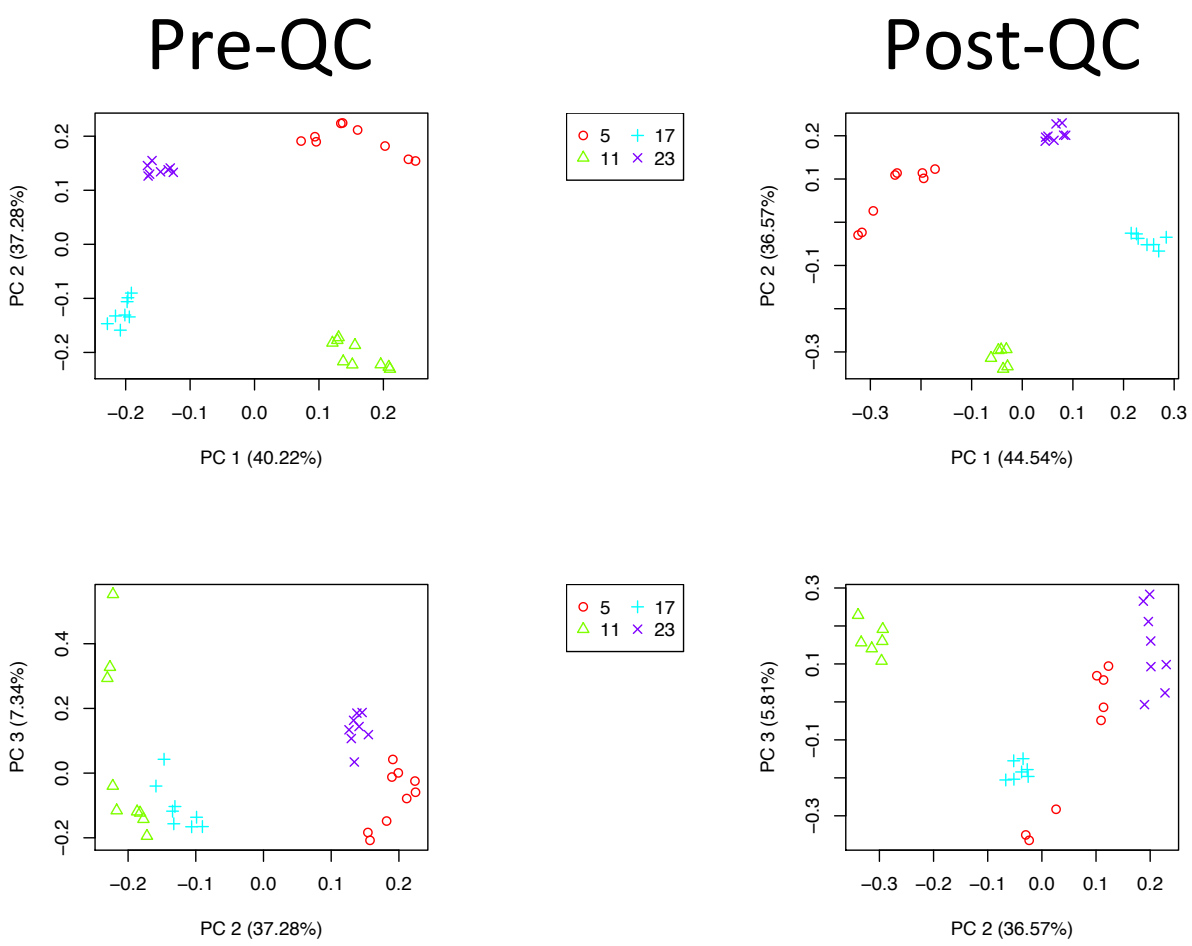


Figure 81. Principal component analysis before and after QC of libraries. The PCA clustering of samples before and after QC filtering shows that the 11:00 time point has improved clustering after removal of low-quality samples. The other conditions show fairly consistent clustering.

4.3.2.3 Differential Gene Expression Analysis

The Trinity RNA-seq analysis platform also includes tools for differential gene expression (DGE) analysis. Several methods are supported by Trinity, including voom (*359*), edgeR (*360*), and DESeq2 (*361*). Which of these tools should be used is still an open question and active area of research (*362*). The kallisto quantification of the *B. braunii* v2.1 transcripts was analyzed for DGE with voom at different combinations of statistical significance (i.e. P-value) and minimum fold-change (FC) thresholds (Table 37). With the loosest parameters (P < 1e-2 and > 1-FC), there were 13,903 differentially expressed transcripts, or 67% of all transcripts. The minimum FC threshold clearly has a stronger impact than the P-value on the detected set of differentially expressed transcripts. When the thresholds are P < 1e-2 and > 16-FC, there were 1,033 differentially expressed transcripts (5% of total). The two settings show starkly different patterns of hierarchical clustering, with the > 16-FC threshold showing more clearly the coexpression patterns (Figure 65). In contrast, the sample correlation matrix is essentially consistent between the two thresholds (Figure 66). In the hierarchical tree of rows in the differential gene expression heatmap, there are clearly several groups of coexpressed transcripts, which can be extracted by cutting the tree into sub-clusters. The next phase of data analysis involves looking in greater detail at the clusters of differentially expressed genes.

P-value	>1-FC	>2-FC	>4-FC	>8-FC	>16-FC
<1e-2	13,903	6,490	3,007	1,768	1,033
<1e-3	12,353	6,310	2,977	1,760	1,027
<1e-4	10,633	5,992	2,897	1,741	1,011
<1e-5	8,754	5,500	2,773	1,693	989
<1e-6	6,890	4,886	2,570	1,605	941
<1e-7	5,184	4,096	2,271	1,467	861
<1e-8	3,600	3,146	1,859	1,212	719
<1e-9	2,363	2,227	1,429	939	552

Table 35. Number of differentially expressed transcripts. These data show that selection of both the p-value threshold and the fold-change threshold will have strong impacts on the number of differentially expressed genes detected from the quantification data. It is not immediately clear from these data what are the optimal thresholds for analysis.

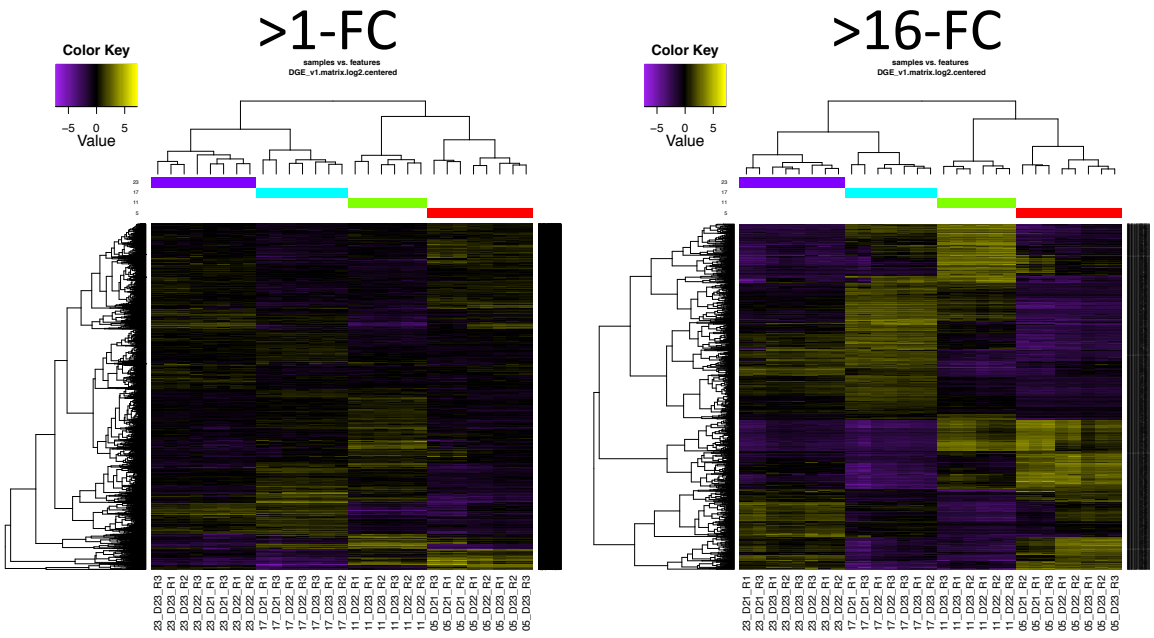


Figure 82. Gene expression heatmap of differentially expressed genes at P < 1e-2 and >1-FC or >16-FC. These data show that utilizing a higher fold-change threshold results in clearer patterns of gene expression. This is important for improving the signal-to-noise ratio when looking for gene co-expression modules.

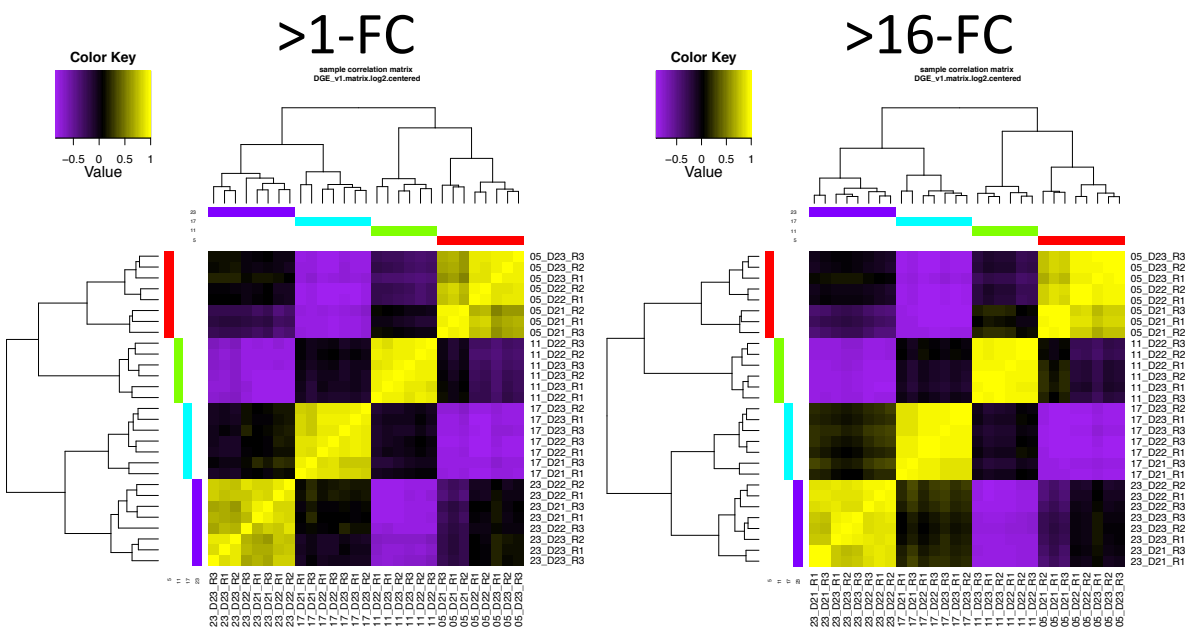


Figure 83. Sample correlation matrix of differentially expressed genes at P < 1e-2 and >1-FC or >16-FC. The samples show better correlation when a high fold-change threshold is applied for selection of differentially expressed genes from the quantification data. This provides further support for utilizing a stringent fold-change threshold.

4.3.2.4 Coexpression of Genes and Functions

The Trinity toolkit offers a few methods for obtaining clusters of coexpressed genes, mainly through either hierarchical or k-means clustering, or a combination of the two. The recommended method is to extract sub-clusters from the tree of hierarchically clustered genes by cutting branches at a certain percentage of maximum tree height. In order to test the effect of the FC threshold on cluster formation, different thresholds were used to generate clusters by cutting at different points along the hierarchical trees (Table 38). There were at maximum 116 and at minimum 3 clusters detected. By cutting at 40% of tree height with a minimum 1-FC threshold, 15 clusters were extracted, 11 of which had more than 50 genes (Figure 67). Some of these clusters show an average gene expression pattern that essentially does not change with time of day, indicating that the low FC threshold allows a great deal of noise into the set of differentially expressed genes. In contrast, when cutting at 40% of tree height with a minimum 16-FC threshold, 13 clusters were extracted, 10 of which had more than 50 genes (Figure 68). The higher FC threshold yields much more consistent clusters with greater power to resolve patterns of differential expression. Cutting the trees at 60% of maximum height for each threshold shows similar results with a smaller number of clusters (Figures 69 and 70). While cutting at a higher percentage of tree height results in a smaller number of clusters, some of the resolution is lost. Therefore it remains an open question as to what is the best method to generate clusters of coexpressed genes. In addition to the tools offered by Trinity, many other programs have been designed to analyze expression data for determination of coexpressed networks of genes (*363*).

In order to examine the functional annotations associated with differentially expressed (DE) genes, functions were counted for each of the 13 distinct clusters of coexpressed genes from cuting at 40% of tree height with >16-FC (Figure 71). Additionally, a column was created with

counts for each of the functions in the set of non-DE genes. The different functional ontologies showed strikingly different patterns of distribution in the differentially expressed clusters and the non-DE genes. Many of the functions detected in the clusters, were also detected (often at a higher frequency) in the non-DE genes, especially for the EC, GO, and PFAM annotations. The KEGG annotations were more unique in the clusters, with less overlap occurring between the clusters and the non-DE genes. These data highlight the challenge of comparing across functional ontologies, but also demonstrate the power of this analytical approach to revealing patterns of transcriptional regulation for various biological processes. Although some of the functions were also found to occur in non-DE genes, it is notable that each cluster contains a set of functions that is not shared with any other cluster. Thus each cluster could represent "functional knobs" that allow the cell to finely tune specific pathways, modulating them either up or down, with varying degrees of "ground level" activity from the non-DE genes. A great deal of highly detailed physiological information could be obtained from careful analysis of the functions found in each cluster. Furthermore, it may be possible to identify the regulatory factors governing the transcriptional flux of each cluster. It is likely that different transcriptional activators and circadian circuits are responsible for each cluster. Comparison of genomic DNA upstream of the transcription start sites (i.e. promoter regions) could yield DNA motifs that are bound by transcription activators or repressors. However, this effort could be complicated by other regulatory and epigenetic factors, such as chromatin condensation, DNA methylation, histone modification, etc.

Table 36. Number of clusters from cutting at different points along hierarchical tree. These data show how clusters are consolidated by cutting at different tree heights. The table also shows how different fold-change thresholds impact the clusters that are cut from the tree.

Percent Height	P < 1e-2 C > 1-FC	P < 1e-2 C > 2-FC	P < 1e-2 C > 4-FC	P < 1e-2 C > 8-FC	P < 1e-2 C > 16-FC
20%	116	116	104	82	64
30%	37	34	35	27	24
40%	15	14	16	12	13
50%	9	10	10	7	7
60%	6	6	7	6	5
70%	6	5	5	5	5
80%	3	3	3	3	3

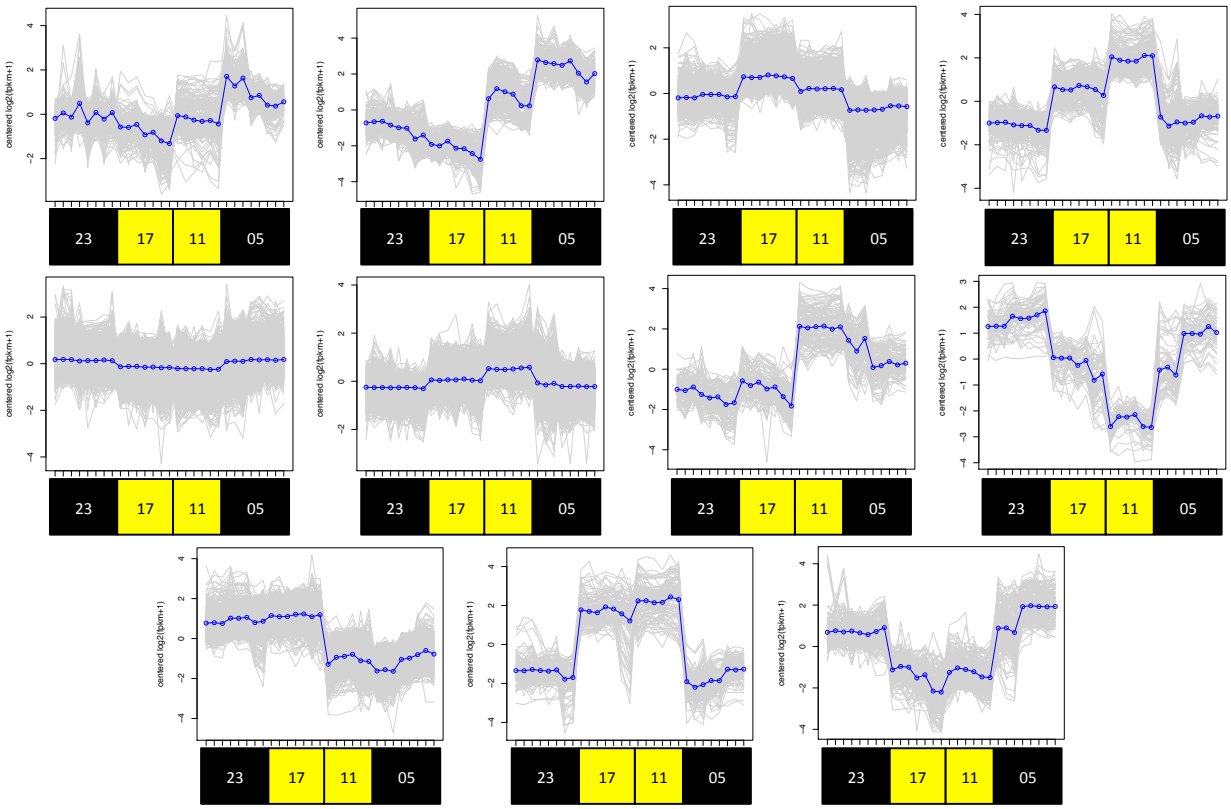


Figure 84. Clusters of genes with >1-FC cut at 40% of tree height. Using a low fold-change threshold results in some highly noisy clusters that do not show any apparent changes in gene expression per time of day. However, there are also good clusters present with distinctive patterns of expression.

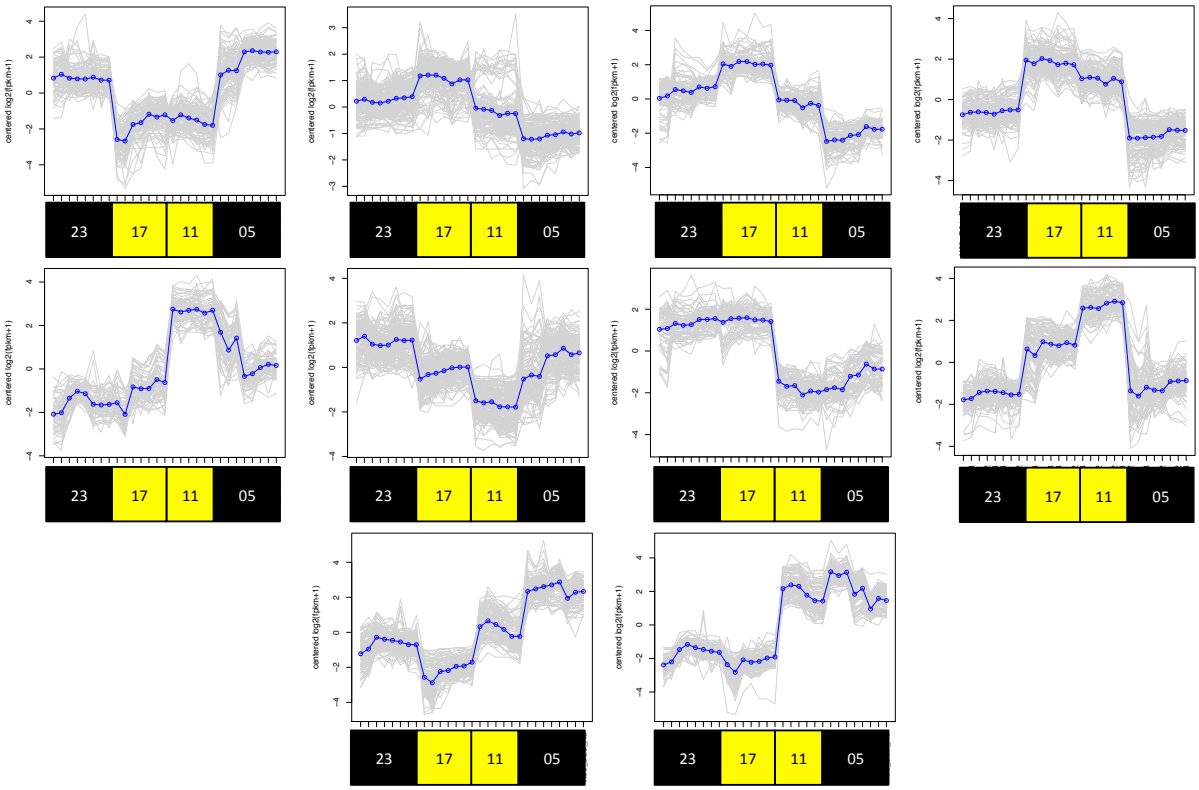


Figure 85. Clusters of genes with >16-FC cut at 40% of tree height. Using a strict fold-change threshold greatly improves the clarity of the clusters that are cut from the hierarchical tree. However, it is possible that there are false negatives (i.e. missing data) due to the stringency of the threshold.

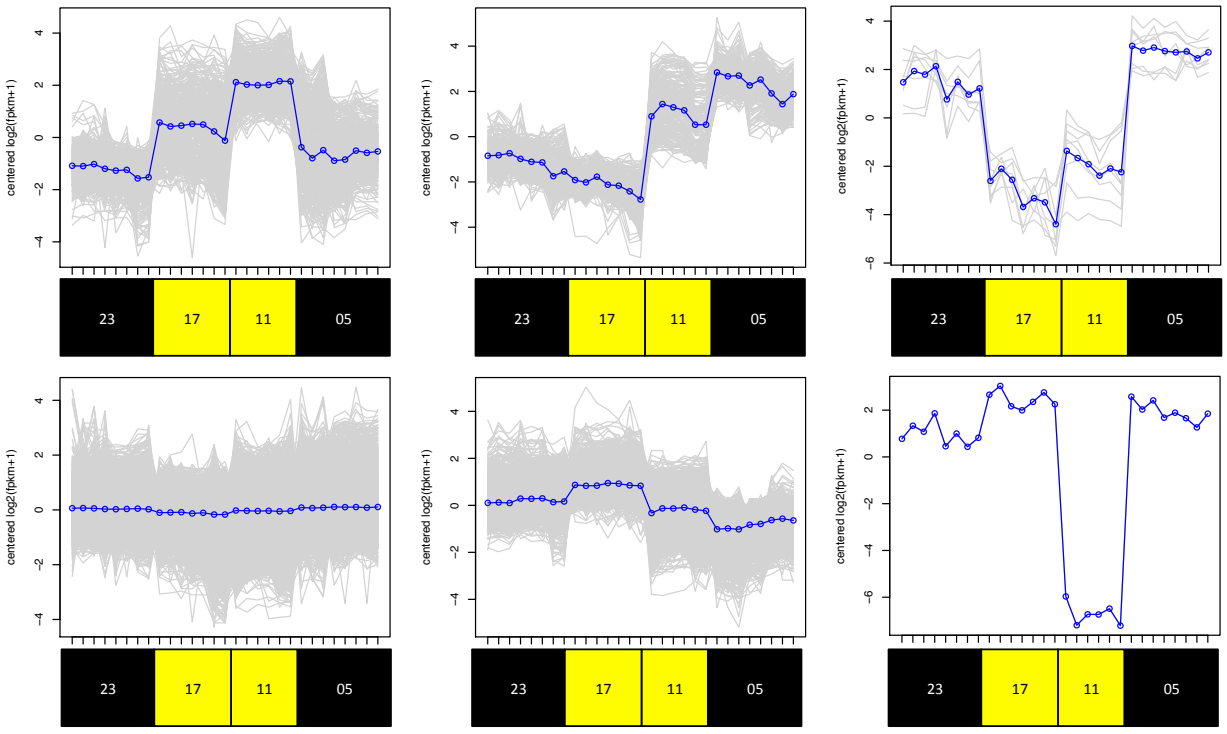


Figure 86. Clusters of genes with >1-FC cut at 60% of tree height. Raising the percentage of tree height at which the tree is cut only worsens the noise, especially when a low fold-change threshold is utilized for selecting differentially expressed genes. Meaningful clusters are not obtained with this approach.

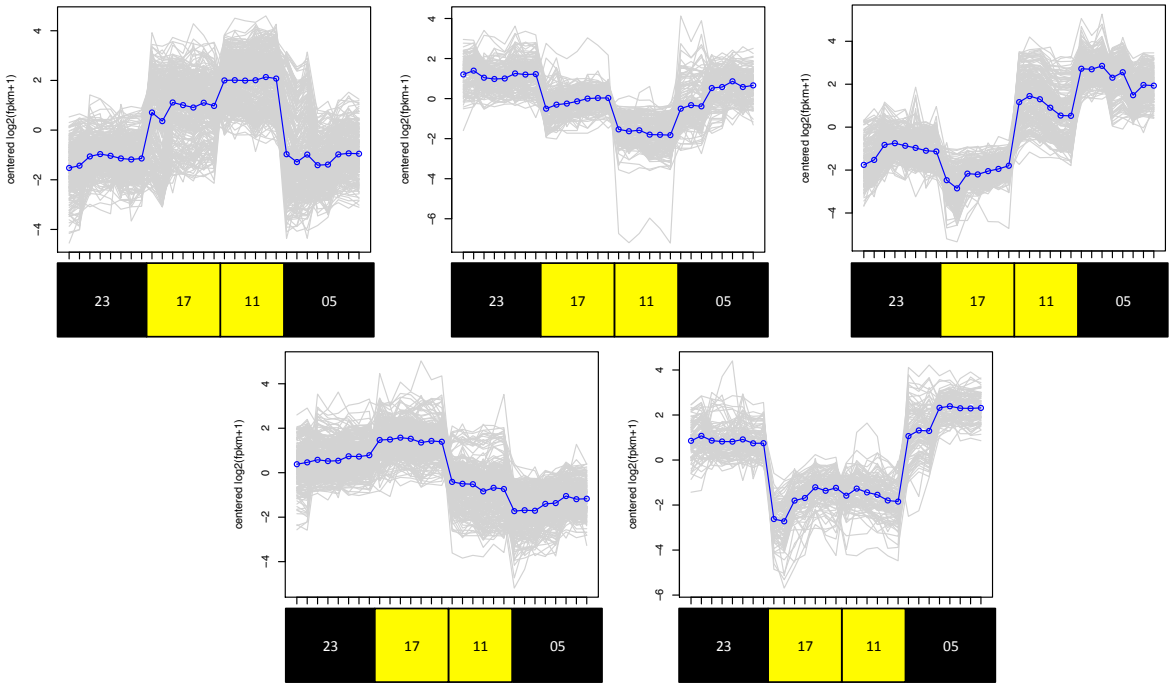


Figure 87. Clusters of genes with >16-FC cut at 60% of tree height. While a higher fold-change threshold improves the signal-to-noise ratio, it is clear from these data that cutting too high on the tree can still weaken the clarity of the clusters obtained from the tree. The parameters must allow for the sufficient differentiation of clusters.

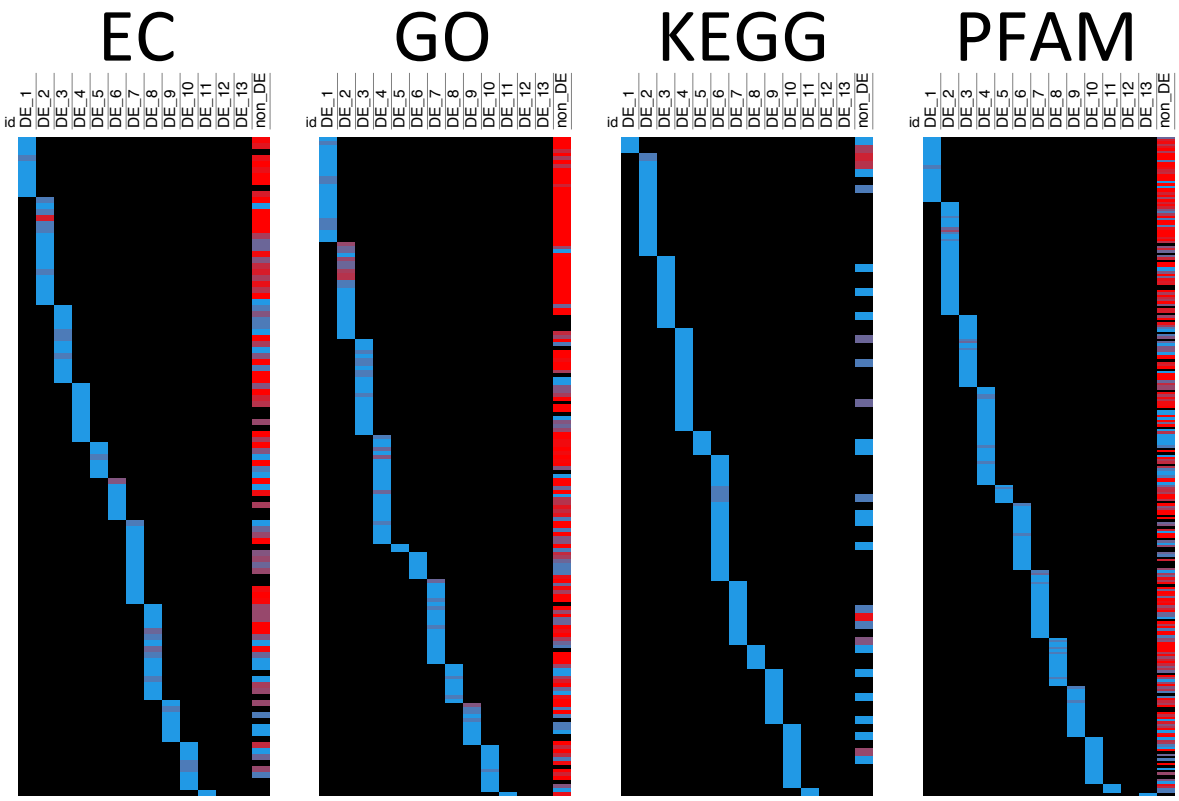


Figure 88. Functional partitioning in clusters of genes with >16-FC cut at 40% of tree height. These data show that each cluster has a distinctive functional signature. There is not any apparent functional overlap between the differentially expressed clusters of genes. However, there is substantial functional overlap of clusters and non-differentially expressed genes.

4.3.2.5 Transcription in Different Key Pathways

Another method for elucidating important patterns of transcriptional regulation along the diel cycle is to examine specific pathways. KEGG provides a powerful tool for conducting this analysis, with the ability to map KEGG orthology terms to defined pathways. Thus the *B. braunii* v2.1 genome annotations were utilized to extract genes within various key pathways and examine their expression patterns in the experiment. For each pathway, the expression data were log-transformed and then converted to Z-scores and visualized with Morpheus. The mean log-transformed expression value for each row was plotted alongside the data, and the columns were ordered by time of day, with intra-group ordering from day 21 to day 23. The pathways examined were protein synthesis and degradation (Figure 72), core transcriptional machinery (Figure 73), DNA replication and cell division (Figure 74), photosynthesis and carbon fixation (Figure 75), and central energy and carbon metabolism (Figure 76). These pathways are in fact each combinations of several pre-defined KEGG pathways. The protein synthesis and degradation pathway consists of aminoacyl-tRNA biosynthesis (ko00970), ribosome (ko03010), proteasome (ko03050), and ubiquitin-mediated proteolysis (ko04120). The core transcriptional machinery pathway consists of RNA polymerase (ko03020), basal transcription factors (ko03022), and spliceosome (ko03040). The DNA replication and cell division pathway consists of DNA replication (ko03030), homologous recombination (ko03440), cell cycle (ko04111), and meiosis (ko04113). The photosynthesis and carbon fixation pathway consists of photosynthesis (ko00195), photosynthesis antenna proteins (ko00196), porphyrin and chlorophyll metabolism (ko00860), and carbon fixation (ko00710). The central energy and carbon metabolism pathway consists of oxidative phosphorylation (ko00190), TCA cycle (ko00020), fatty acid biosynthesis (ko00061), and terpenoid backbone biosynthesis (ko00900). Using the KEGG framework enables a powerful

analytical approach to broadly classify genes according to their higher-level biological functions. This approach could be further expanded to look at more functions in the gene expression dataset for *B. braunii*, as there are many more pre-defined KEGG pathways available that were not included in this analysis. Of the pathways examined so far, protein synthesis and degradation and core transcriptional machinery both show little to no association of expression patterns with time of day. The DNA replication and cell division and central energy and carbon metabolism pathways both show a moderate degree of expression periodicity in association with time of day. Unsurprisingly, photosynthesis and carbon fixation showed the strongest signature of gene expression regulation according to time of day.

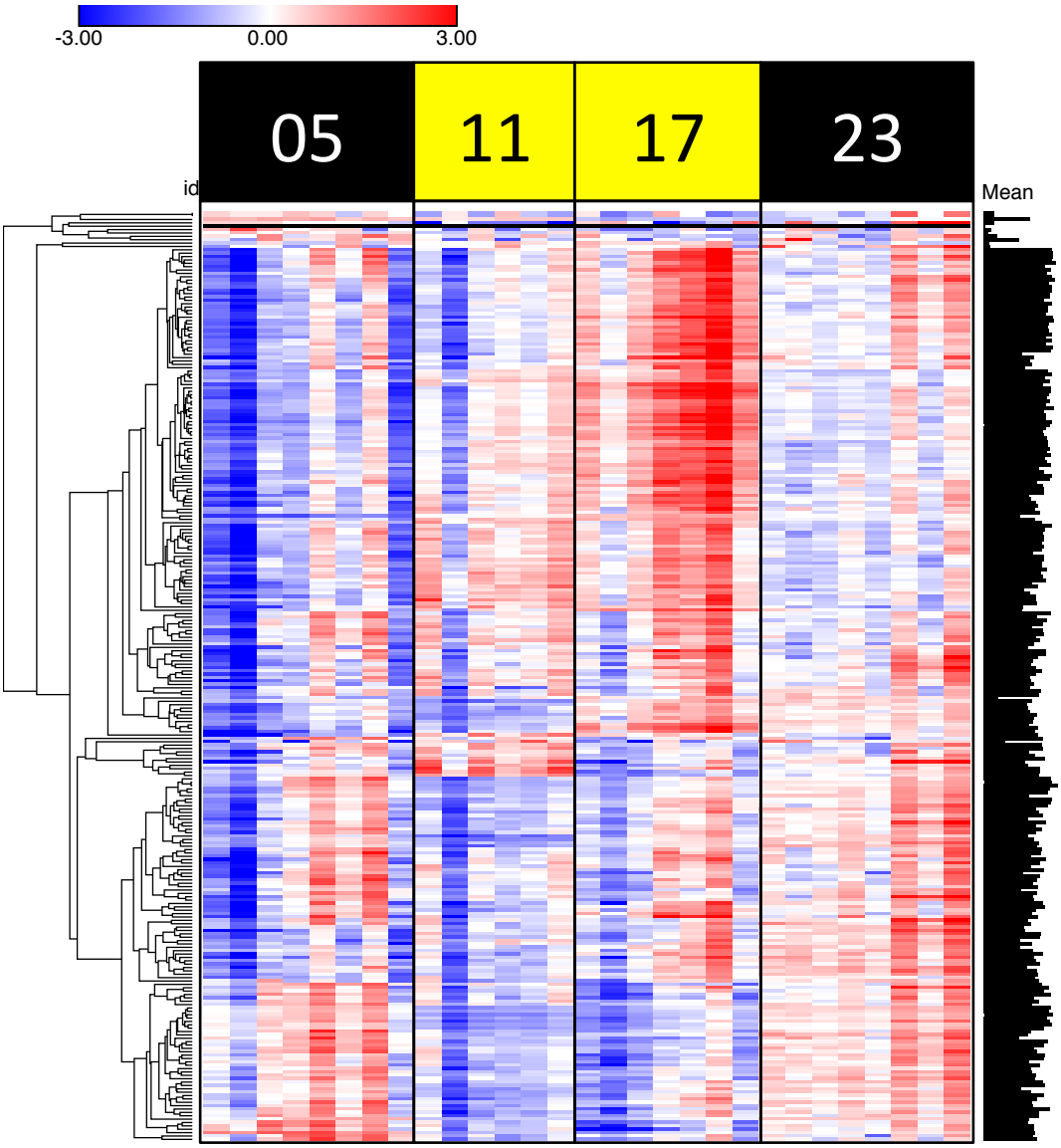


Figure 89. Expression patterns of protein synthesis and degradation. These data show that a number of genes in the pathway are strongly expressed at the 17:00 time point. However, there is clearly some day-to-day variation, especially evident in the 5:00 and 23:00 time points.

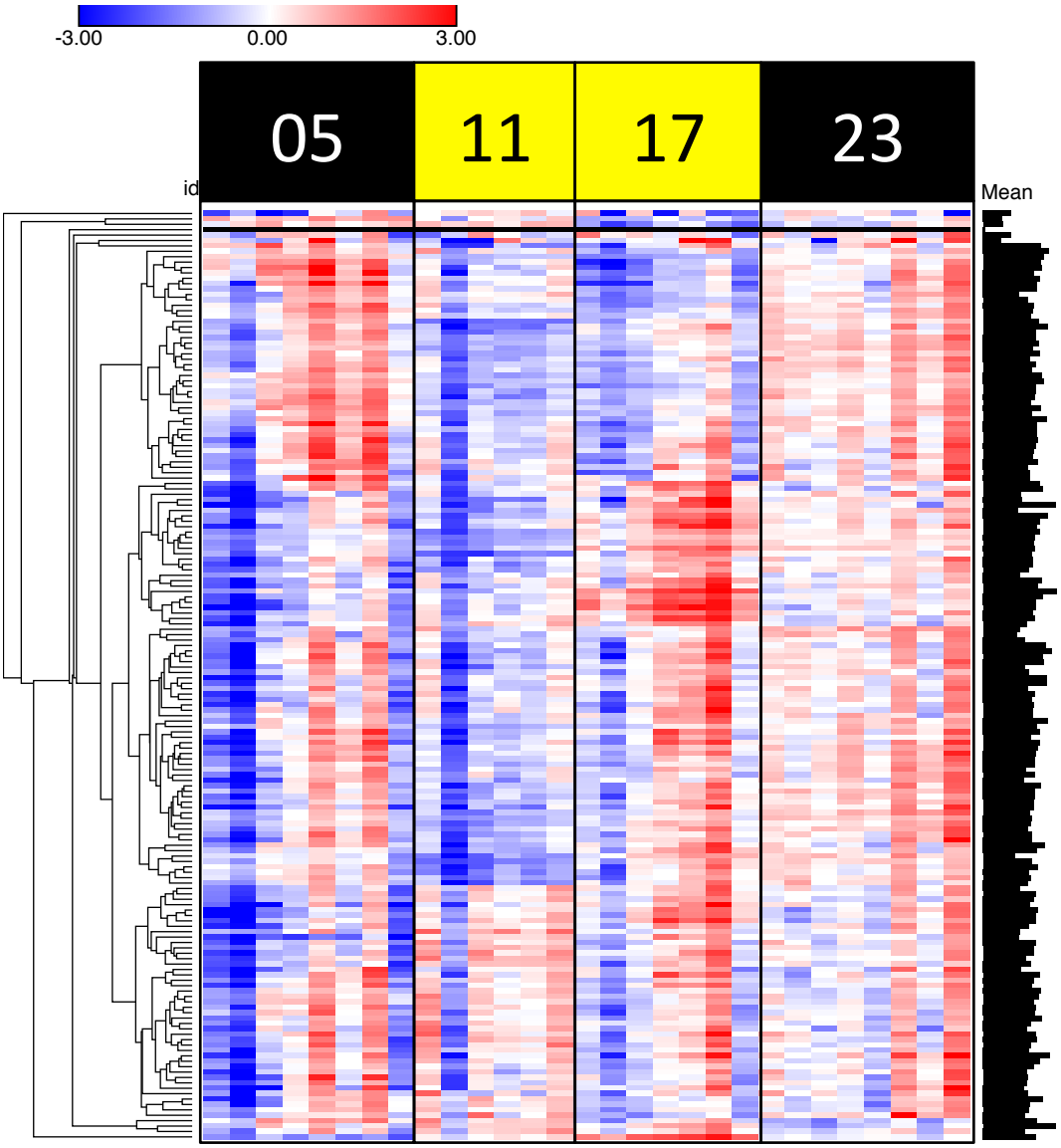


Figure 90. Expression patterns of core transcriptional machinery. These data show consistent upregulation of the genes in the pathway at the 23:00 time point. However, there is again clearly day-to-day variation, especially at the 5:00 and 17:00 time points.

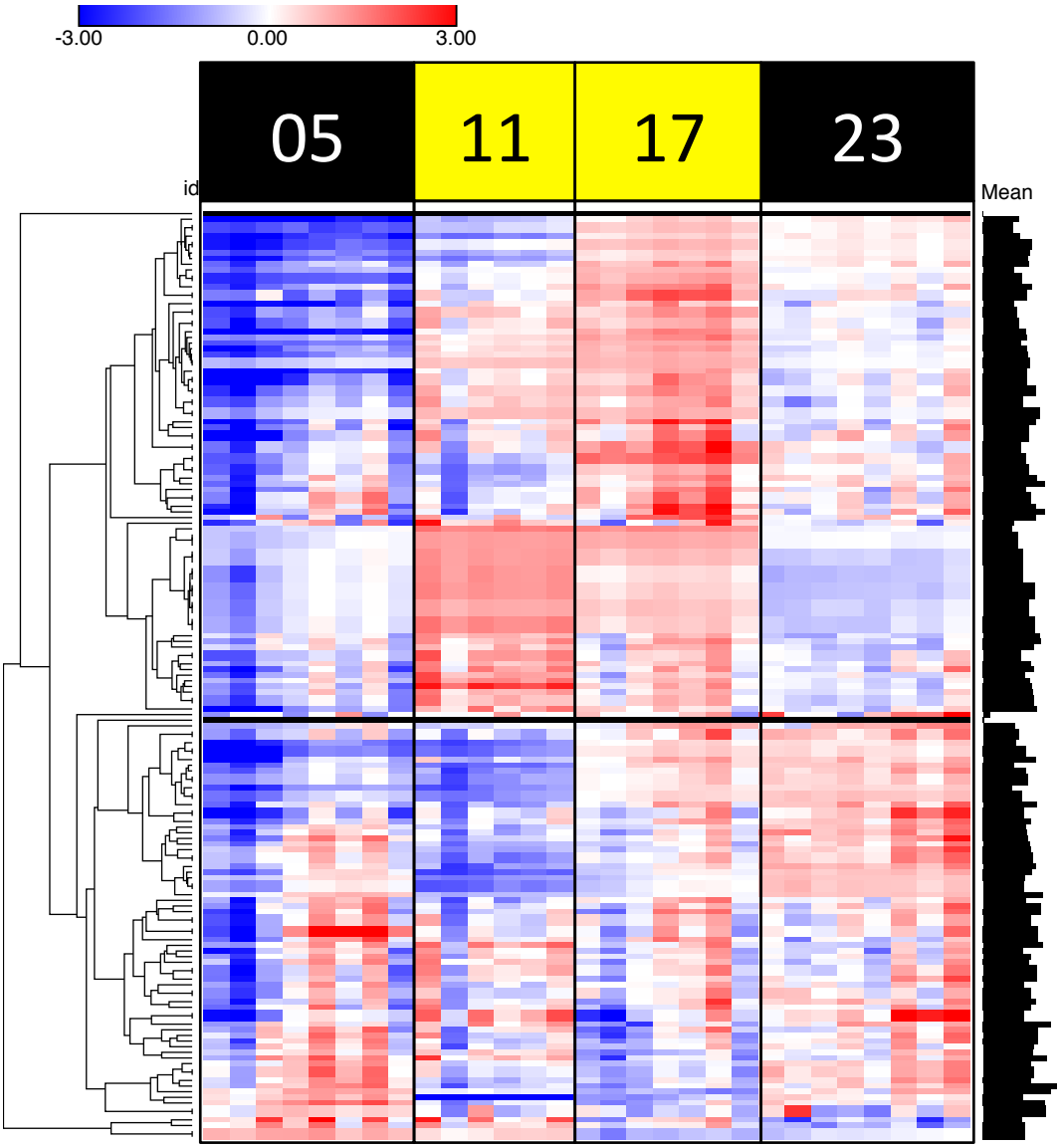


Figure 91. Expression patterns of DNA replication and cell division. The genes in this pathway mostly show consistent expression patterns across the days, with some exceptions. Notably, there is a clear upregulation of some genes during the 11:00 and 17:00 time points, collected in the light. This indicates that light has a positive impact on DNA replication and cell division processes.

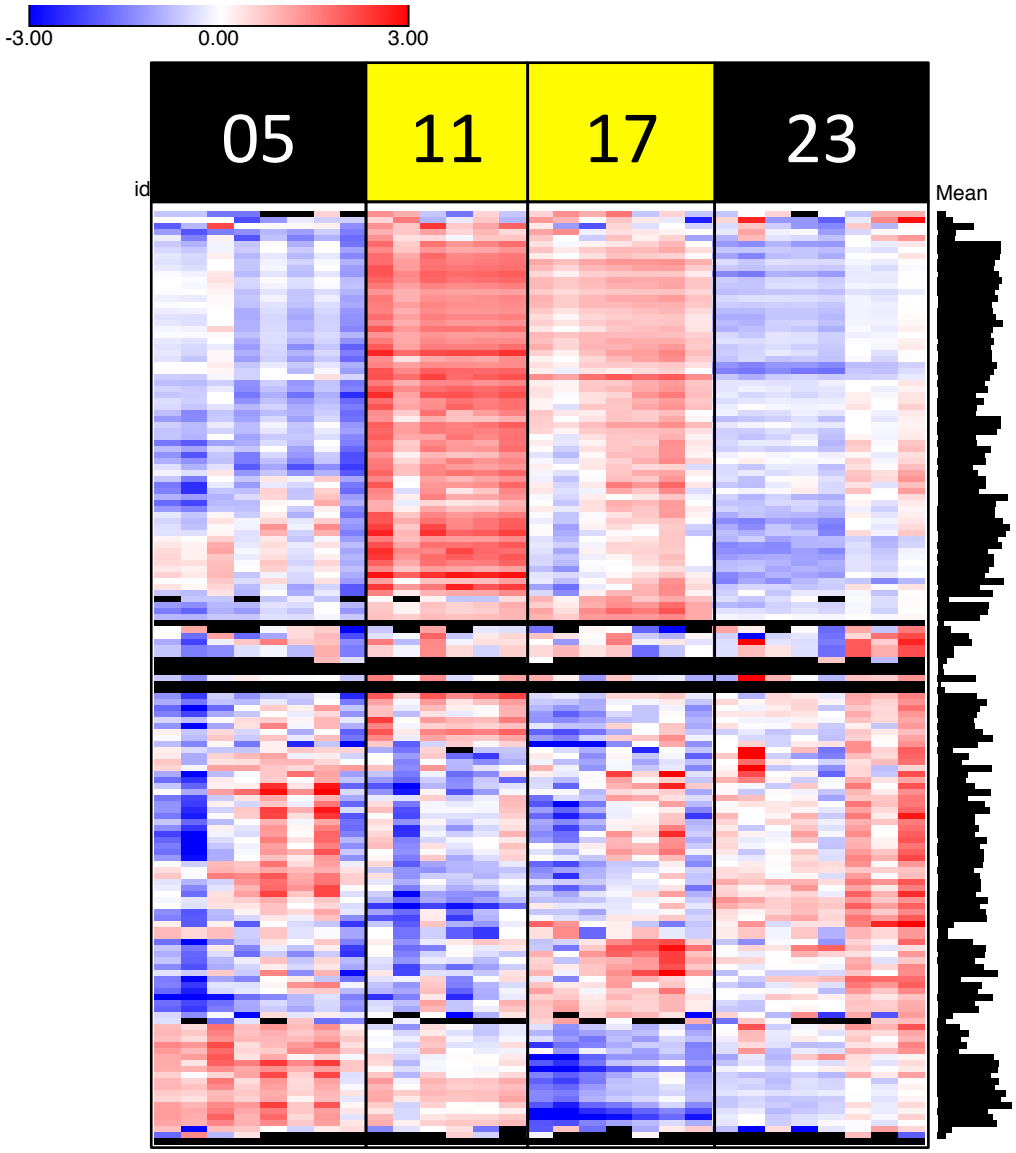


Figure 92. Expression patterns of photosynthesis and carbon fixation. These data should that many photosynthesis genes are upregulated under light conditions, at the 11:00 and 17:00 time points, as would be expected. Interestingly, there is also a sub-set of genes that appears to oscillate independently of light conditions. There are also several predicted genes that do not appear to have any expression at all under the experimental conditions.

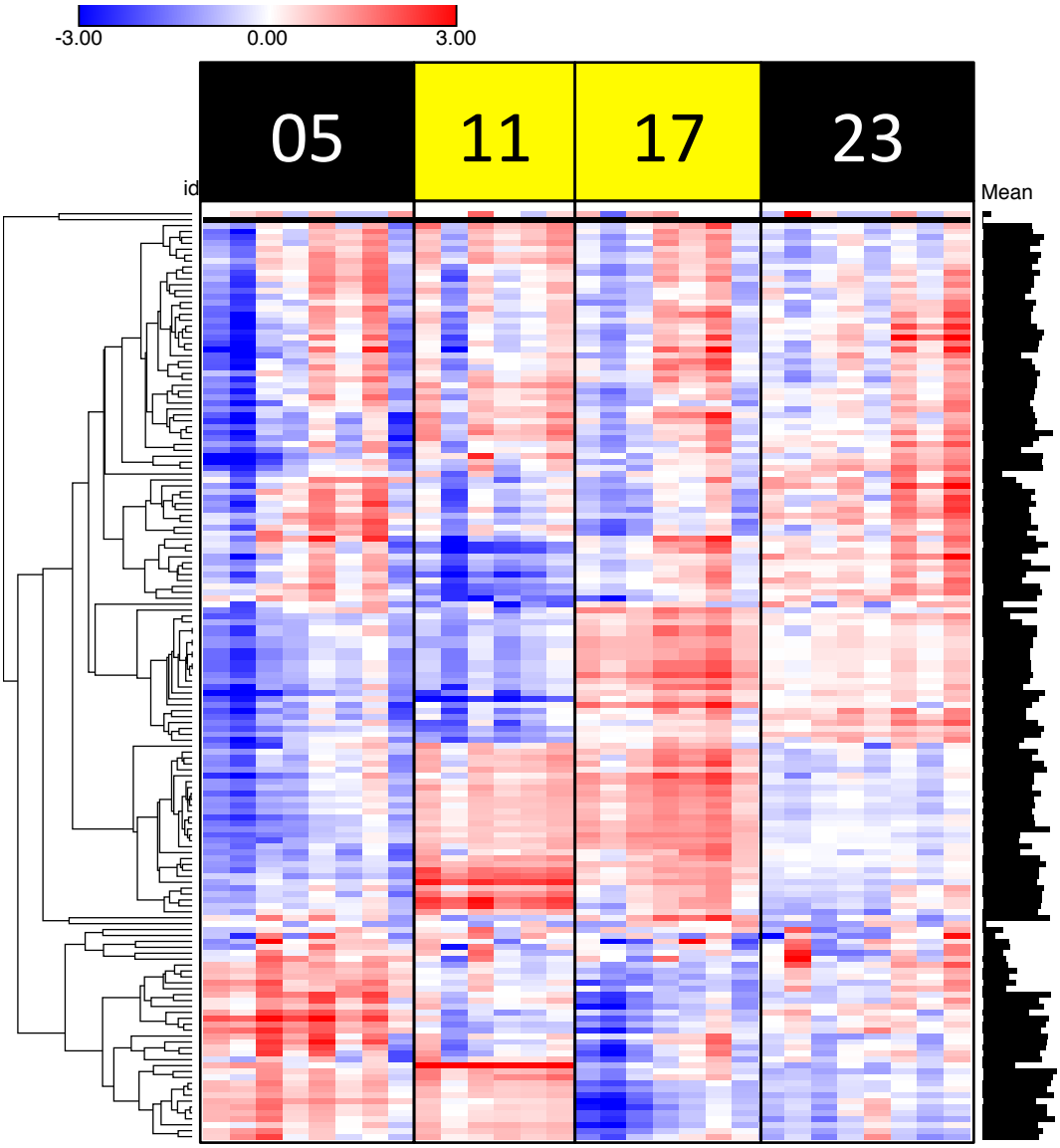


Figure 93. Expression patterns of central energy and carbon metabolism. These data show that the central energy and carbon metabolism pathways have quite varied expression patterns, with light-independent switching on and off, as well as light-dependent upregulation, and apparently time-independent genes.

4.3.3 Analysis of Metabolite Profile

Polar and nonpolar metabolites were separately extracted from the *B. braunii* biomass samples and then analyzed by liquid chromatography electrospray ionization tandem mass spectrometry (LC-ESI-MS/MS) using different columns. Bead beating was employed to disrupt the cell walls and then either ethanol/water or chloroform/methanol solutions were utilized to extract the polar and nonpolar metabolites, respectively. The polar metabolites were separated with a hydrophilic interaction chromatography (HILIC) column, while the nonpolar metabolites were separated with either a C_{18} or a C_{18}-lipid column. The resulting data enabled both targeted and untargeted analyses of the metabolite profile, both of which will be discussed in the following sections.

4.3.3.1 Targeted Analysis of Metabolite Profile

An MS library of polar metabolite standards was used to screen the LC-ESI-MS/MS results from the polar metabolite extracts in order to positively identify compounds in the experimental samples. There were 141 metabolites detected with varying degrees of confidence across all of the *B. braunii* biomass samples (Table 39). Manual inspection of these positively identified polar metabolites enabled classification of each metabolite into one of eight categories (Figure 77). The specified categories were amino acids, nucleic acids, lipids, sugars, coenzymes, hormones, small metabolites, and large metabolites. There were 44 amino acids and related intermediates identified in the samples, almost twice the number of any other category. The nucleic acids and small metabolites categories each had just over 20 compounds. Interestingly, some of the nucleic acids identified were methylated or otherwise modified nucleobases, supporting the presence of epigenetic DNA base modifications in *B. braunii*. Similarly, the amino acids category

included modified amino acids supporting the presence of post-translational modification mechanisms. However, inspection of the raw targeted polar metabolomics data revealed that some of the detected metabolites were very weakly supported (Figure 78). For example, some of the metabolites were also detected in the blanks and some were detected in only a handful of the experimental samples. In order to reduce low-quality signal, a simple filter heuristic was developed and applied to the data. If the average MS peak height of the blanks was greater than 10% of the average peak height of the experimental samples, or if the metabolite was not detected in one or more of the experimental samples, then the metabolite was discarded. This filtration method reduced the set of targeted polar metabolites from 141 to 92 detected compounds.

Unfortunately, a comparable MS library of nonpolar metabolites for targeted analysis of the data does not currently exist; therefore the nonpolar data could not be broadly screened to identify compounds. However, several triterpenoids were purified from *B. braunii* by Tatli *et al (364)*, enabling a small-scale screen of the nonpolar data using these compounds as standards (Figure 79). Five isomers of botryococcene were sent to the JGI for analysis and all of them were positively identified in the nonpolar metabolomics data. Hierarchical clustering of the samples showed good consistency among the replicates but did not reveal any patterns of metabolite flux in accordance with time of day. Purification of other lipids from *B. braunii* could be tremendously useful for elucidating specific compounds in the nonpolar metabolomics dataset. This is important for gaining deeper insight into the metabolism of fatty acids and terpenoids in *B. braunii*. As it currently stands, there is essentially no information available about the specific nonpolar compounds that *B. braunii* is producing.

In order to test for time of day changes in the positively identified polar metabolites, the MS peak heights of each metabolite in each sample were log-transformed and then converted to

Z-scores and visualized with Morpheus (Figure 80). The columns were grouped by time of day, from day 21 to 23, and the rows were hierarchically clustered according the Spearman rank correlation. The mean log-transformed signal was calculated for each row. This analysis revealed no patterns of change in metabolite pool in association with the time of day. Hierarchical clustering of the columns did not improve the results, as the columns did not cluster according to time of day (data not shown). However, there are patterns of change in the metabolite pool that appear to associate with the day of collection. This indicates that while metabolite pools may not change with time of day, they are shifting over time.

Table 37. Polar metabolites detected in targeted analysis. This table lists all of the metabolites detected in the targeted polar analysis. There is a total of 141 compounds that were detected in the experiment, although only 92 of these metabolites are detected with high confidence.

Identified Polar Metabolites		
1-aminocyclopropane-1-carboxylate	d-(+)-galactosamine	leucine
1-methyladenosine	d-(+)-glucosamine	lumichrome
1-methylnicotinamide	d-(+)-raffinose	maltose (peak1)
2-amino-2-methylpropanoate	d-(+)-trehalose	mannitol
2-aminoisobutyric acid	d-lactose	methionine
2-hydroxypyridine	d-mannosamine	methyl acetoacetate
2-methylpropanal oxime	d-pantothenic acid	n-acetyl-D-mannosamine
2'-deoxyadenosine	d-sorbitol	n-acetyl-L-methionine
3-aminoisobutanoate	deoxycarnitine	n-acetyl-L-leucine
3-methoxytyramine	deoxycytidine	n-acetyl-L-phenylalanine
4-aminobutanoate	deoxycytidine	n-methyl-D-aspartic acid
4-guanidinobutanoate	diethanolamine	n-methyl-L-glutamate
4-hydroxy-2-quinolinecarboxylic acid	ectoine	nalpha-acetyl-L-lysine
4-hydroxy-L-phenylglycine	folic acid	nicotinamide
4-hydroxy-L-proline	galactitol	nicotinic acid (niacin)
4-imidazoleacetic acid	glutathione	norleucine
5-aminolevulinic acid	glycerol 2-phosphate	o-acetyl-L-carnitine
5-aminovaleric acid	glycine	ophthalmic acid
5-hydroxymethyluracil	guanine	phosphocholine
5-methylcytosine hydrocloride	guanosine 3'5'-cyclic monophosphate	phosphocreatine
5'-deoxyadenosine	homoserine	pipecolate
5'-methylthioadenosine	hypoxanthine	proline

6-hydroxynicotinate	indole-3-acetamide	pyridoxal
abscisic acid	inosine	pyridoxine
acetylcholine	isocitrate	retinoate
adenine	isonicotinic acid	ribitol
adenosine	L-alanine	riboflavin
adenosine 2'3'-cyclic monophosphate	L-allothreonine	s-(5'-adenosyl)-L-homocysteine
adenosine 5'-monophosphate	L-arabitol	s-(5'-adenosyl)-L-methionine
alpha-aminoadipate	L-arginine	sn-glycero-3-phosphocholine
alpha-d-glucose 1-phosphate	L-aspartate	sn-glycerol 3-phosphate
asparagine	L-carnitine	sucrose
beta-alanine	L-glutamic acid	taurine
betaine	L-glutamine	thiamine
biotin	L-histidine	thymine
carnosine	L-histidinol	thyrotropin releasing hormone
choline	L-hydroxyproline (cis-4-hydroxy-d-proline)	trans-4-hydroxyproline
cis-4-hydroxy-d-proline	L-isoleucine	trigonelline
citrate	L-lysine	tryptamine
citrulline	L-norvaline	uridine
cortisol	L-ornithine	uridine-5-monophosphate
cortisol 21-acetate	L-phenylalanine	urocanate
creatine	L-serine	valine
creatinine	L-threonine	vanillin
cytidine	L-tryptophan	xanthine
cytidine 2'3'-cyclic mono-phosphate	L-tyrosine	xylitol
cytosine	lauroylcarnitine	y-aminobutyric acid

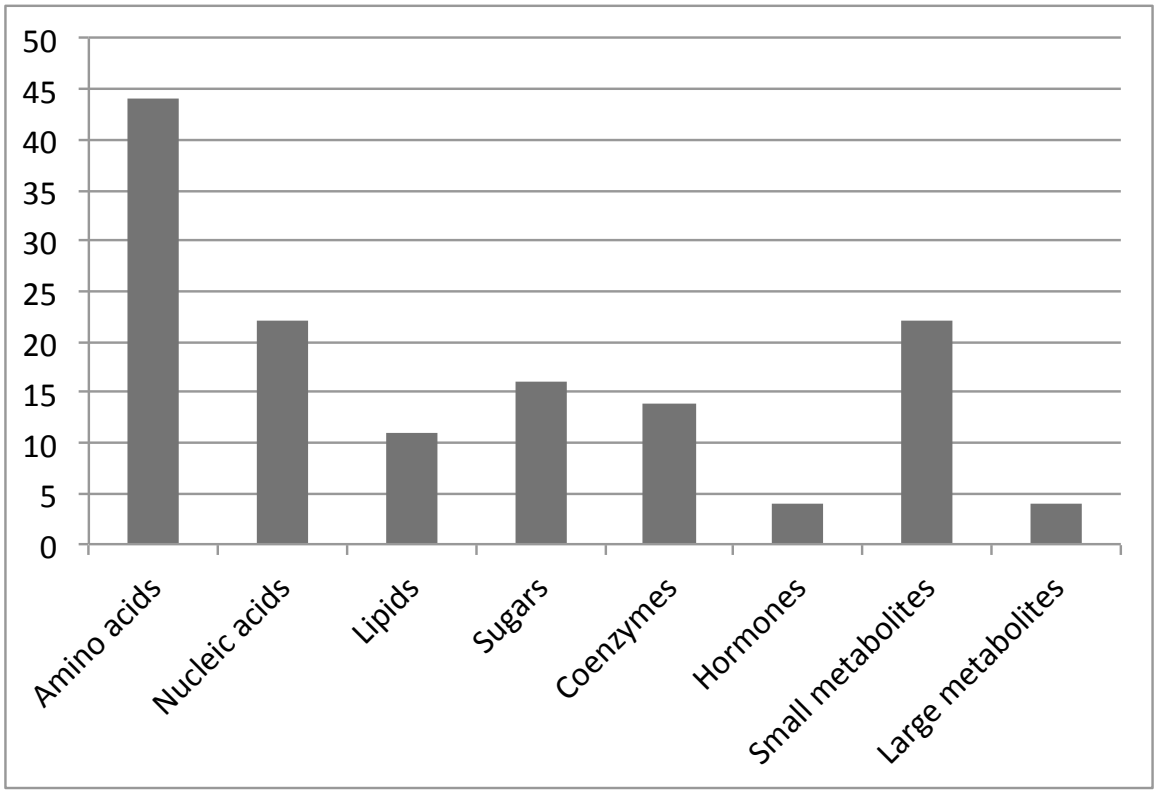


Figure 94. Molecular classification of polar metabolites. The polar metabolites identified in the targeted analysis were manually classified into one of eight categories. The largest category of metabolites identified in the experiment was amino acids, followed by nucleic acids, and then by an assortment of miscellaneous small metabolites.

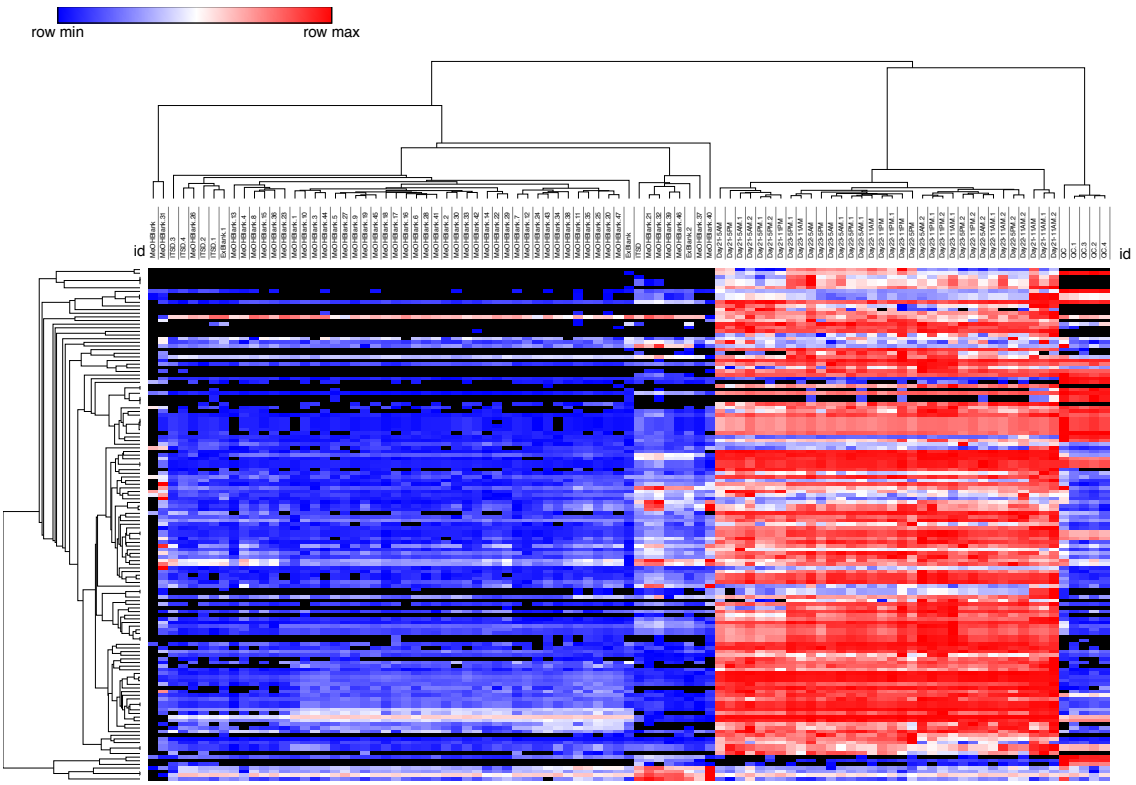


Figure 95. Quality control analysis of targeted polar metabolites. These data show the signals for each of the identified metabolites in the experimental samples in contrast with the controls (i.e. blanks and standards). This table demonstrates that most of the identified polar metabolites have strong detection in the experimental samples.

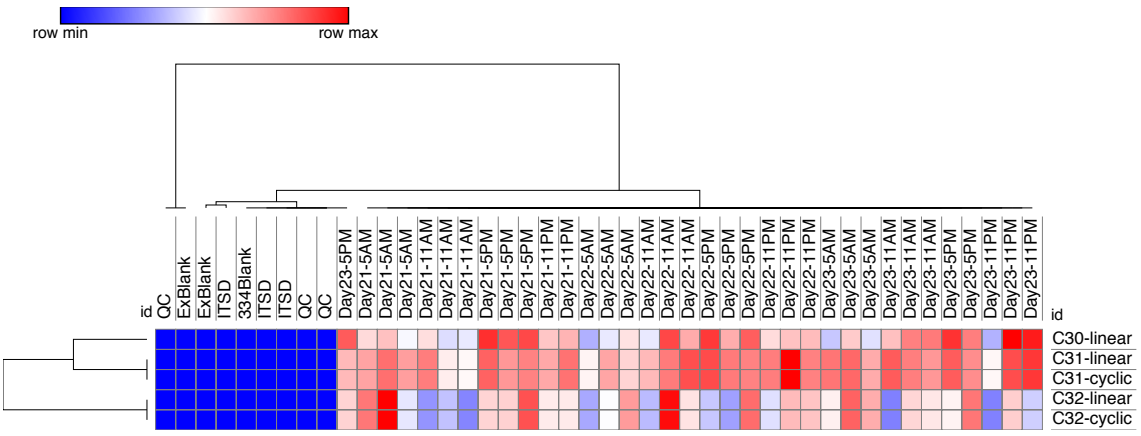


Figure 96. Quality control analysis of botryococcene standards. Targeted analysis of nonpolar metabolites is technically challenging, in part due to a lack of available standards. Since botryococcene standards were in supply, they could be utilized to perform a targeted analysis of these compounds. These data show strong detection for each compound in the experimental samples, as compared against the controls.

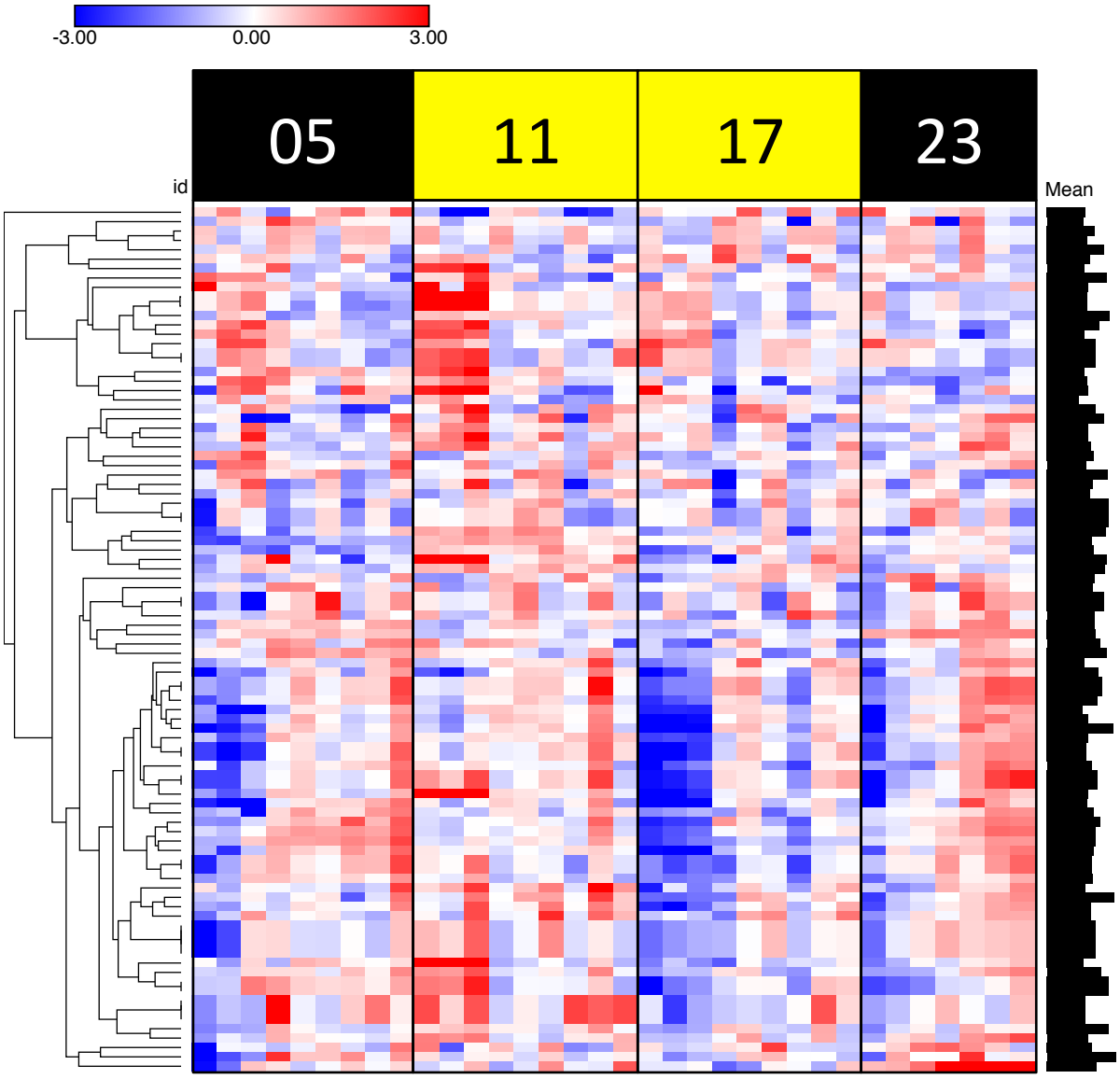


Figure 97. Changes in targeted polar metabolites per time of day. After filtration of low-confidence metabolites, the data were structured according to time of day in order to search for apparent patterns in metabolite profile. These data show that the polar metabolites identified in the targeted analysis do not have any apparent correlation with time of day.

4.3.3.2 Untargeted Analysis of Metabolite Profile

Another possibility is that the small sample of targeted polar metabolites is insufficient to reveal broader patterns of change in the metabolite pools. Therefore, the untargeted metabolomics data were investigated for such changes in metabolite signature associated with time of day. Both the polar and nonpolar untargeted data were analyzed, with the same previously described filtration heuristic applied to the datasets. Four datasets were investigated, polar metabolites detected with positive ion mode (Figure 81), polar metabolites detected with negative ion mode (Figure 82), nonpolar metabolites detected with positive ion mode (Figure 83), and nonpolar metabolites detected with negative ion mode (Figure 84). After filtration, there were 2,714 polar metabolites detected with positive ion mode, 2,105 polar metabolites detected with negative ion mode, 2,698 nonpolar metabolites detected with positive ion mode, and 1,239 nonpolar metabolites detected with negative ion mode. Interestingly, all of the untargeted analyses show results that are consistent with the targeted analysis. That is, there does not appear to be any major groups of metabolites that flux in accordance with the time of day. However, the same pattern of shifting over the course of days appears once again, with fairly stark differences between the day 21 and day 23 samples, and the day 22 samples showing intermediate detection. These data present a strong contrast to the transcriptomics data, which very clearly showed a substantial portion of gene expression fluxing in accordance with time of day.

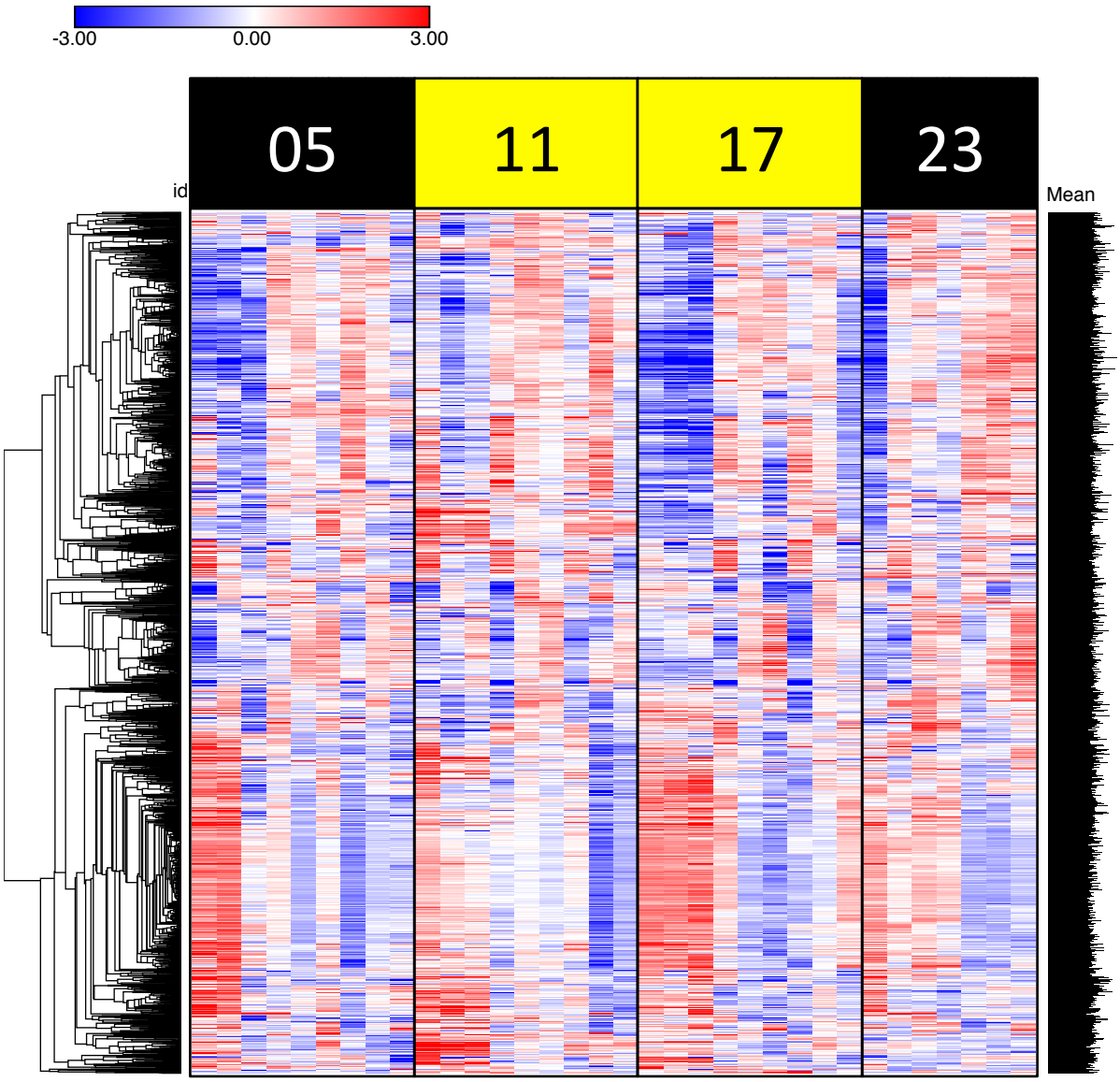


Figure 98. Untargeted analysis of polar metabolites in positive ion mode. These data show that the polar metabolites in the untargeted analysis do not have an apparent correlation with time of day. However, there do appear to be day-to-day variations in the metabolite pool.

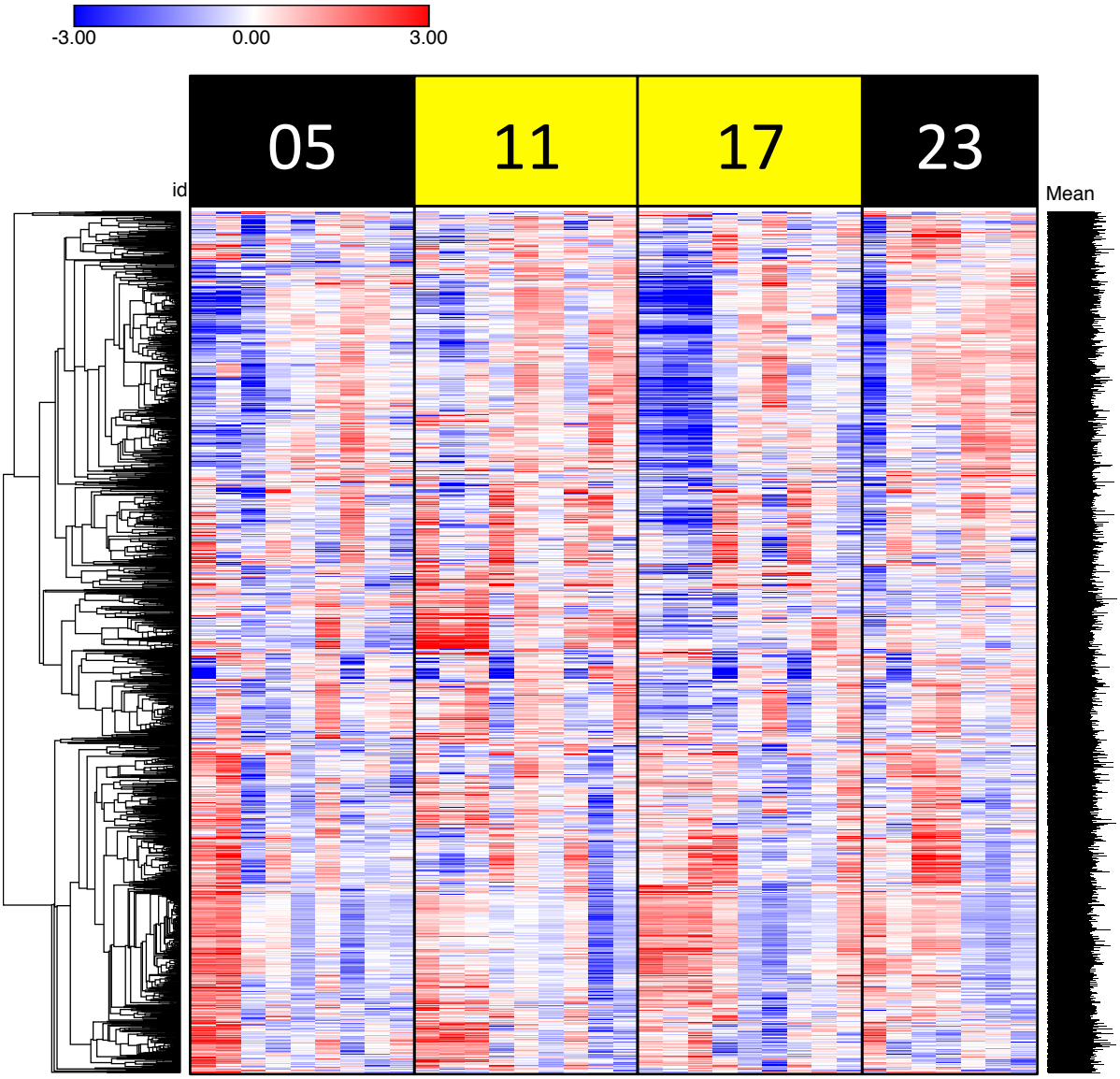


Figure 99. Untargeted analysis of polar metabolites in negative ion mode. These data show that the polar metabolites in the untargeted analysis do not have an apparent correlation with time of day. However, there do appear to be day-to-day variations in the metabolite pool.

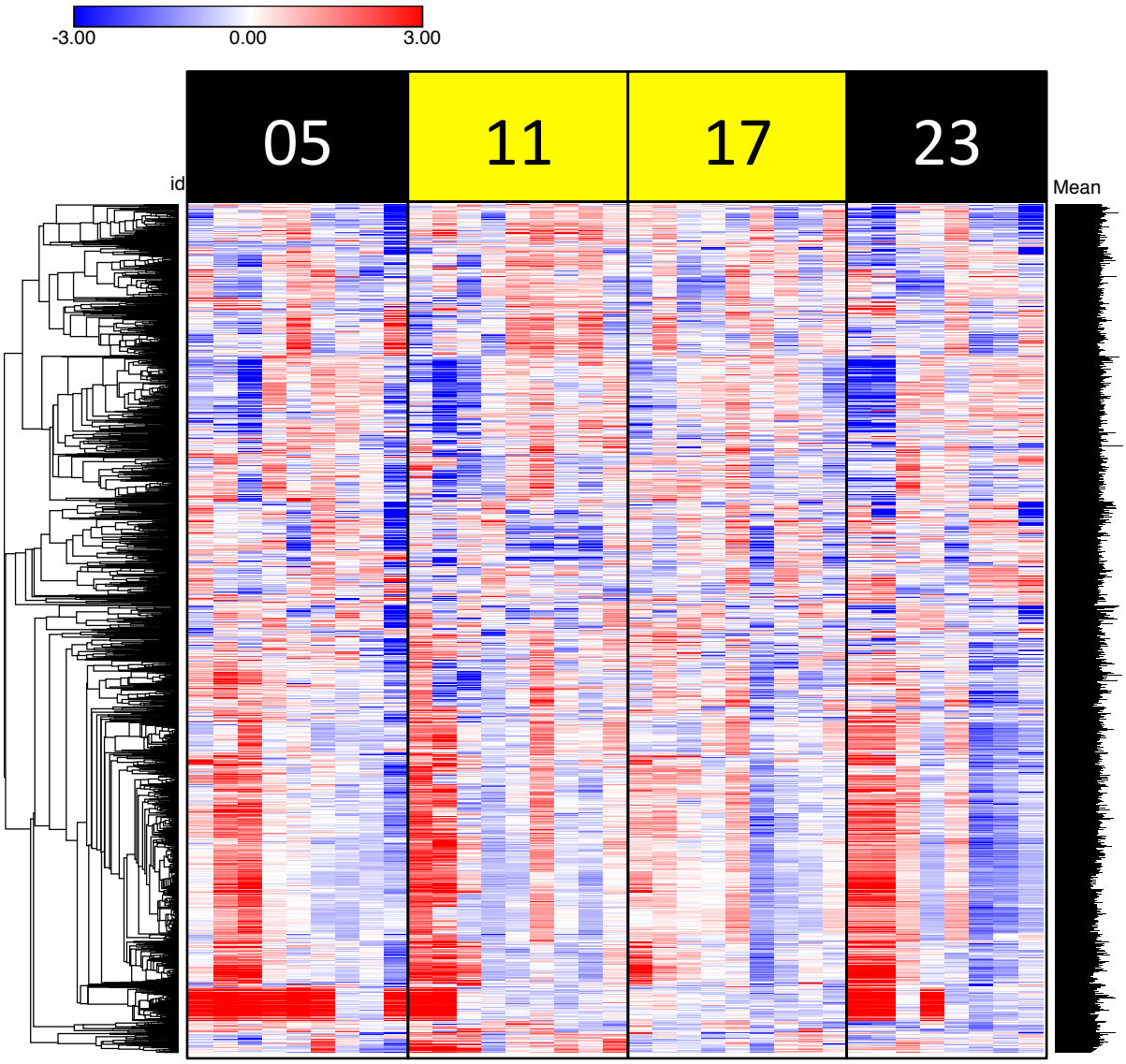


Figure 100. Untargeted analysis of nonpolar metabolites in positive ion mode. These data show that the nonpolar metabolites in the untargeted analysis do not have an apparent correlation with time of day. However, there do appear to be day-to-day variations in the metabolite pool.

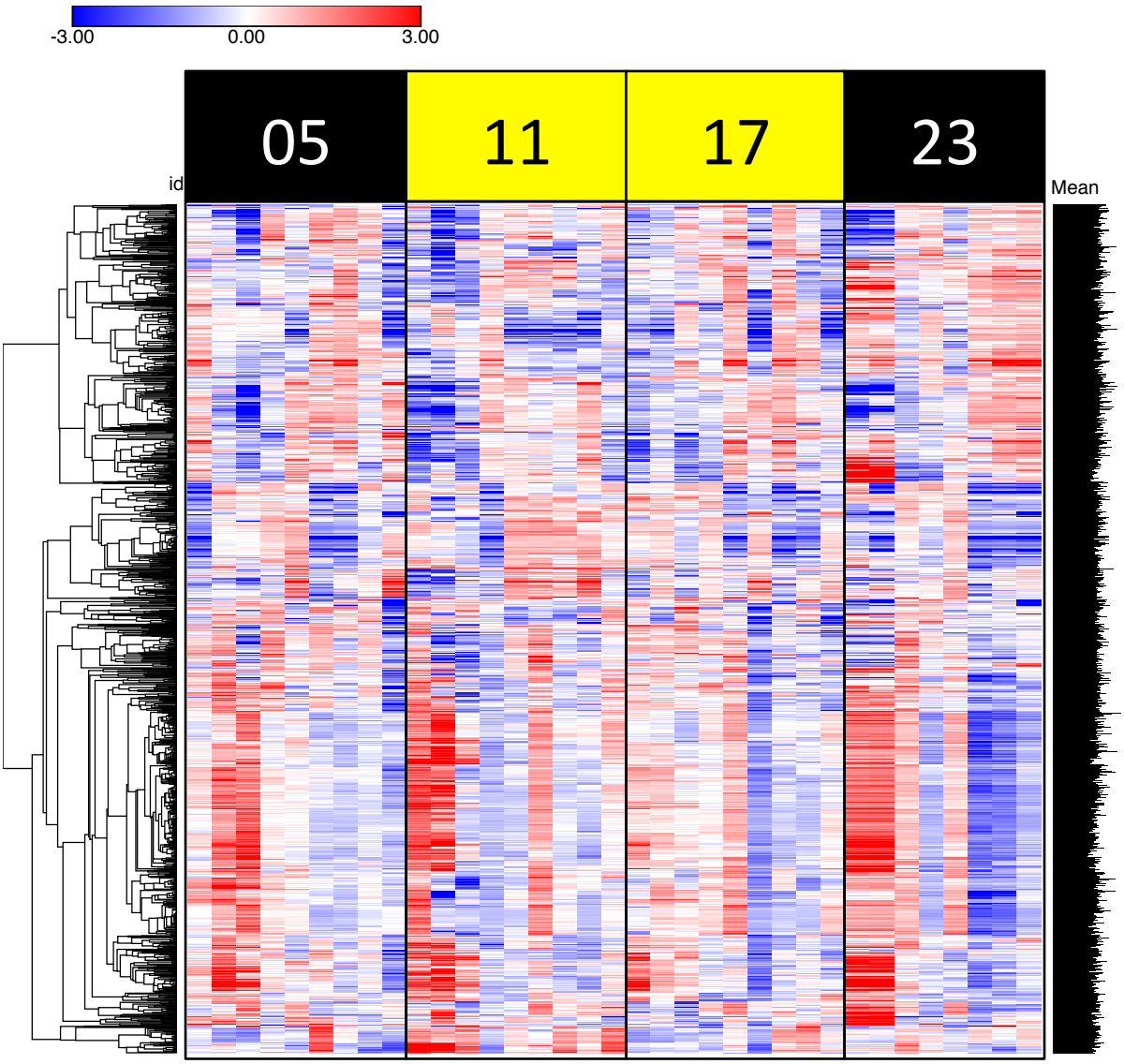


Figure 101. Untargeted analysis of nonpolar metabolites in negative ion mode. These data show that the nonpolar metabolites in the untargeted analysis do not have an apparent correlation with time of day. However, there do appear to be day-to-day variations in the metabolite pool.

4.4 Conclusion

This work has provided ground-breaking analysis of gene expression and metabolite profile in *B. braunii* race B (Showa). The scale of the data created in this study is unprecedented for the field of *B. braunii* research. The results have given enormous amounts of insight into the naturally occurring rhythms of gene expression in the species. This provides a deeper understanding of the basic physiological and metabolic processes that are important. Moreover, the RNA sequencing data are a useful, publicly available resource for anyone studying *B. braunii*. The data can be broadly used to study many different processes in greater detail. The analyses presented in this work only scratch the surface of the possibilities. Future work could include deeper analyses of specific processes in *B. braunii* and comparison with well-studied models.

The metabolomics data are particularly special and promise to reveal a great deal of information about metabolic functions in *B. braunii*. However, the field of metabolomics is still quite new and the availability of tools to process and analyze the data are lacking. As developers continue to create new tools and analytical platforms, more meaningful information can be extracted from the metabolomics datasets generated in this work. There is the possibility of reconstructing nearly the entire set of metabolic pathways operating in the species. New statistical methods, specially designed for metabolomics data, could help improve the signal-to-noise ratio and extract sub-sets of metabolites with specific patterns of flux.

5. CONCLUSION

Based on the work presented in this dissertation, the following section aims to achieve two goals: 1) summarize and discuss the key findings of the work, and 2) look to the future and evaluate the direction of research. The specific fields under consideration are genome sequencing and assembly, Viridiplantae evolution, metabolism and physiology in *B. braunii*, and finally sustainability and biotechnology.

5.1 Genome Sequencing and Assembly

Technologies for genome sequencing and assembly are both rapidly evolving, which makes this area of research incredibly exciting. New platforms from companies like Oxford Nanopore, 10X Genomics, Dovetail Genomics, BioNano Genomics are adding great value and new opportunities to improve on assemblies built from the traditional Illumina and PacBio platforms. Algorithms for assembly are growing to take advantage of increased processing capabilities, such as larger numbers of parallel CPUs arrayed in supercomputing clusters, co-processing technologies such as Phi, and powerful GPUs. However, there is still a lot of room for growth in the fundamental theories of genome assembly. Much of the underlying theory was developed in the late 20th century, and all of these new technologies are challenging some of the basic assumptions, such as Lander-Waterman coverage theory (*365*).

Despite the prevalence of DBG-based genome assemblers, there is surprisingly little fundamental research available on k-mer distributions in natural genomic sequences. A better basic understanding of k-mers could help yield improvements in DBG-based assembly algorithms. For example, identifying and filtering out k-mers that are unlikely to be true genomic sequences, or improving path-finding approaches in DBG-based genome assembly graphs. Error correction of

reads is another application of k-mer theory that could yield improvements in genomic models. Alternatively, there may be entirely new approaches to genome assembly. One major area of interest is the utilization of machine learning algorithms. Such algorithms could be used to identify and remove contaminating sequences from complex eukaryotic assemblies (*366*). They could also potentially be implemented to achieve entirely new assembly algorithms, alongside many other applications in downstream genomic analyses (*367*).

The *B. braunii* genome stands to benefit enormously from such advances in sequencing and assembly technologies. Unlike certain model organisms such as *A. thaliana* or *C. reinhardtii*, the *B. braunii* genome has received very little attention. Yet it has great potential to open up new insights into fundamental biological processes that govern lipid metabolism and secretion. Moreover, the colony structure of *B. braunii* is highly unique in that it consists of multiple cells embedded in a shared environment. While individual cells can be isolated from the colony, they are unable to survive and propagate (*368*). Thus although *B. braunii* is technically a single-celled organism, in that it does not appear to have any type of tissue differentiation, there is clearly an element of multicellular coordination in the formation and maintenance of the colony structure. Further study of the colony propagation process and intra-colony cellular interactions could help reveal deeper insights into the evolution of multicellular organisms. Similar analyses were attempted using a comparative genomics approach with the *C. reinhardtii* and *V. carteri* genomes, when those were some of the only green algal genomes available (*369*). While *V. carteri* is a true multicellular organism with different cell types forming a spheroid, and *C. reinhardtii* is a true unicellular organism with motile single cells, *B. braunii* represents a kind of intermediate between the two. Therefore there is a strong impetus to continue refining the *B. braunii* genome with the latest advances in sequencing and assembly technologies.

5.2 Gene Evolution in Viridiplantae

More broadly, improvements in genome sequencing and assembly can be applied to the entire Viridiplantae clade, the major barriers being funding and manpower. The work presented in this dissertation has already clearly shown the power of comparative analyses including the full scope of Viridiplantae genomes. Originally, the comparative genomics analyses were limited to the green algae clade (Chlorophyta), but curiosity drove the inclusion of all Viridiplantae genomes available from the Phytozome database. This extension turned out to be enormously important, revealing distinctive genomic signatures across the Viridiplantae. In particular, the analyses consistently showed the expansion of certain functions in the Embryophyta as compared against the Chlorophyta. In another light, this revealed the basic functions of Chlorophyta, which formed the functional foundations that enabled expansion in Embryophyta. Had the analysis been limited to the Chlorophyta, an enormous amount of information would have been missed.

Perhaps one of the most significant outcomes of this work is the application of KEGG pathways to the Viridiplantae genomes, which has never before been so comprehensively evaluated. This work clearly demonstrated the utility of KEGG orthology and pathways, and implemented a simple term counting approach to determine species-specific and clade-specific genomic signatures on an unprecedented scale. The KEGG pathways in particular provide different lenses to view gene evolution in all of these species. It is clear that different pathways have undergone different selective pressures, with some pathways highly conserved at low gene copy numbers, while other pathways have genes that were duplicated many times. While this is not exactly new information, the analytical approach presented here is new, and provides a proof of concept that other researchers can build upon in the future.

5.3 *B. braunii* Metabolism and Physiology

While a genome assembly is useful for understanding the repertoire of biological functions present in an organism, further information is needed to understand the active interpretation of genomic sequences. In this respect, the transcriptomics and metabolomics data for *B. braunii* developed and presented in this dissertation are unprecedented in the field of *B. braunii* research. Although transcriptome data has been generated previously for *B. braunii*, never has it been prepared on such a massive scale and with such a strong experimental design. The biological replicates built into the diel experiment presented here were essential for obtaining statistical confidence in the patterns of differential gene expression. The metabolomics data generated in this work represents the first of its kind for *B. braunii* and is a significant milestone in the field of *B. braunii* research.

These datasets put *B. braunii* in a position to serve as a model organism for systems biology analyses that integrate multiple "omics" datasets. For example, the construction of genome-scale metabolic models will facilitate interpretation of the metabolomics data and enable more detailed studies of metabolic flux. Although the metabolite profile does not appear to shift with time of day in *B. braunii*, it is possible that flux through the metabolite pools does change with time of day, but not the overall pool size. Unfortunately, the diel experiment was not designed to capture information about metabolic flux, which would require isotopic labeling and pulse-chase experiments with biomass collected at the different times of day. Such experiments are possible and could be pursued in the future if sufficient funding and manpower become available. In contrast, the gene expression profile showed clear patterns of change in accordance with time of day, supporting circadian regulation of cellular processes in *B. braunii*.

5.4 Sustainability and Biotechnology

Not only is *B. braunii* an excellent model for systems biology analyses of fundamental biological processes, it is potentially a valuable source of information for "bioprospecting" of industrially relevant genes and enzymes. The biotechnology sector is rapidly growing and plays an important role in the broader economy of the United States of America (*370*). The development of sustainable biotechnology is essential for the continuation of human civilization. Reliance on fossil deposits of petroleum presents a critical strategic vulnerability, in the sense that they are non-renewable resources and are geographically limited. These resources will eventually be depleted and it is imperative that new technologies are developed now, before the limits of fossil resources begin to negatively impact the global economy. The development of an advanced bioeconomy requires a deeper understanding of basic biological processes. Such knowledge will enable researchers to engineer new solutions to meet critical societal needs (*371*). Algae in particular could play a significant role in the development and deployment of sustainable biotechnology platforms (*372*). Numerous commodities could be obtained from algae, and more broadly photosynthetic microbes, enabling a new paradigm of primary production in human civilization (*373*). Given the role of *B. braunii* in the formation of petroleum deposits throughout geological time, it would be quite fitting for this species to play a leading role in transforming the ecosystem of industrial materials and energy production. With the new insights presented in this dissertation, *B. braunii* takes one small step closer to achieving this goal.

REFERENCES

1. T. Tashiro *et al.*, Early trace of life from 3.95 Ga sedimentary rocks in Labrador, Canada. *Nature* **549**, 516-518 (2017).
2. M. S. Dodd *et al.*, Evidence for early life in Earth's oldest hydrothermal vent precipitates. *Nature* **543**, 60-64 (2017).
3. M. Lliros *et al.*, Pelagic photoferrotrophy and iron cycling in a modern ferruginous basin. *Scientific reports* **5**, 13803 (2015).
4. A. Tagliabue *et al.*, The integral role of iron in ocean biogeochemistry. *Nature* **543**, 51-59 (2017).
5. M. D. Brasier, J. Antcliffe, M. Saunders, D. Wacey, Changing the picture of Earth's earliest fossils (3.5-1.9 Ga) with new approaches and new discoveries. *Proc Natl Acad Sci U S A* **112**, 4859-4864 (2015).
6. G. Wachtershauser, From volcanic origins of chemoautotrophic life to Bacteria, Archaea and Eukarya. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* **361**, 1787-1806; discussion 1806-1788 (2006).
7. G. Wachtershauser, Groundworks for an Evolutionary Biochemistry: the Iron-Sulphur World. *Progress in biophysics and molecular biology* **58**, 85-201 (1992).
8. G. Wachtershauser, Before Enzymes and Templates: Theory of Surface Metabolism. *Microbiological Reviews* **52**, 452-484 (1988).
9. J. P. Dworkin, A. Lazcano, S. L. Miller, The roads to and from the RNA world. *Journal of Theoretical Biology* **222**, 127-134 (2003).

10. L. E. Orgel, Prebiotic chemistry and the origin of the RNA world. *Crit Rev Biochem Mol Biol* **39**, 99-123 (2004).

11. M. P. Callahan *et al.*, Carbonaceous meteorites contain a wide range of extraterrestrial nucleobases. *Proc Natl Acad Sci U S A* **108**, 13995-13998 (2011).

12. S. L. Miller, A Production of Amino Acids Under Possible Primitive Earth Conditions. *Science* **117**, 528-529 (1953).

13. J. L. Bada, New insights into prebiotic chemistry from Stanley Miller's spark discharge experiments. *Chemical Society reviews* **42**, 2186-2196 (2013).

14. K. Zahnle, L. Schaefer, B. Fegley, Earth's earliest atmospheres. *Cold Spring Harbor perspectives in biology* **2**, a004895 (2010).

15. M. A. Keller, D. Kampjut, S. A. Harrison, M. Ralser, Sulfate radicals enable a non-enzymatic Krebs cycle precursor. *Nat Ecol Evol* **1**, (2017).

16. G. Wachtershauser, The place of RNA in the origin and early evolution of the genetic machinery. *Life (Basel)* **4**, 1050-1091 (2014).

17. B. H. Patel, C. Percivalle, D. J. Ritson, C. D. Duffy, J. D. Sutherland, Common origins of RNA, protein and lipid precursors in a cyanosulfidic protometabolism. *Nat Chem* **7**, 301-307 (2015).

18. T. Czaran, B. Konnyu, E. Szathmary, Metabolically Coupled Replicator Systems: Overview of an RNA-world model concept of prebiotic evolution on mineral surfaces. *J Theor Biol* **381**, 39-54 (2015).

19. J. Pereto, P. Lopez-Garcia, D. Moreira, Ancestral lipid biosynthesis and early membrane evolution. *Trends in biochemical sciences* **29**, 469-477 (2004).

20. P. A. Lindahl, Stepwise Evolution of Nonliving to Living Chemical Systems. *Origins of life and evolution of the biosphere : the journal of the International Society for the Study of the Origin of Life* **34**, 371-389 (2002).

21. K. Zaremba-Niedzwiedzka *et al.*, Asgard archaea illuminate the origin of eukaryotic cellular complexity. *Nature* **541**, 353-358 (2017).

22. C. M. Weisman, S. R. Eddy, Gene Evolution: Getting Something from Nothing. *Current biology : CB* **27**, R661-R663 (2017).

23. S. Sengupta, P. G. Higgs, Pathways of Genetic Code Evolution in Ancient and Modern Organisms. *J Mol Evol* **80**, 229-243 (2015).

24. R. S. Gupta, Impact of genomics on the understanding of microbial evolution and classification: the importance of Darwin's views on classification. *FEMS Microbiol Rev* **40**, 520-553 (2016).

25. M. Parks *et al.*, Ancient population genomics and the study of evolution. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* **370**, 20130381 (2015).

26. M. Lynch *et al.*, Genetic drift, selection and the evolution of the mutation rate. *Nature reviews. Genetics* **17**, 704-714 (2016).

27. E. Benard, S. Lebre, C. J. Michel, Genome evolution by transformation, expansion and contraction (GETEC). *Biosystems* **135**, 15-34 (2015).

28. S. Martinez Cuesta, S. A. Rahman, N. Furnham, J. M. Thornton, The Classification and Evolution of Enzyme Function. *Biophysical journal* **109**, 1082-1086 (2015).

29. O. P. Judson, The energy expansions of evolution. *Nature Ecology & Evolution* **1**, 0138 (2017).

30. N. Nelson, C. F. Yocum, Structure and Function of Photosystems I and II. *Annual review of plant biology* **57**, 521-565 (2006).

31. D. T. Flannery, M. R. Walter, Archean tufted microbial mats and the Great Oxidation Event: new insights into an ancient problem. *Australian Journal of Earth Sciences* **59**, 1-11 (2012).

32. L. M. Ward, J. L. Kirschvink, W. W. Fischer, Timescales of Oxygenation Following the Evolution of Oxygenic Photosynthesis. *Origins of life and evolution of the biosphere : the journal of the International Society for the Study of the Origin of Life* **46**, 51-65 (2016).

33. A. Lazcano, S. L. Miller, How Long Did It Take for Life to Begin and Evolve to Cyanobacteria. *J Mol Evol* **39**, 546-554 (1994).

34. A. G. Tomkins *et al.*, Ancient micrometeorites suggestive of an oxygen-rich Archaean upper atmosphere. *Nature* **533**, 235-238 (2016).

35. R. Frei, C. Gaucher, S. W. Poulton, D. E. Canfield, Fluctuations in Precambrian atmospheric oxygenation recorded by chromium isotopes. *Nature* **461**, 250-253 (2009).

36. A. Y. Mulkidjanian *et al.*, The cyanobacterial genome core and the origin of photosynthesis. *Proc Natl Acad Sci U S A* **103**, 13126-13131 (2006).

37. T. Dagan *et al.*, Genomes of Stigonematalean cyanobacteria (subsection V) and the evolution of oxygenic photosynthesis from prokaryotes to plastids. *Genome biology and evolution* **5**, 31-44 (2013).

38. B. E. Rubin *et al.*, The essential gene set of a photosynthetic organism. *Proc Natl Acad Sci U S A* **112**, E6634-6643 (2015).

39. M. Eisenhut *et al.*, The photorespiratory glycolate metabolism is essential for cyanobacteria and might have been conveyed endosymbiontically to plants. *Proc Natl Acad Sci U S A* **105**, 17199-17204 (2008).

40. N. Kashtan *et al.*, Single-cell genomics reveals hundreds of coexisting subpopulations in wild Prochlorococcus. *Science* **344**, 416-420 (2014).

41. L. Tirichine, C. Bowler, Decoding algal genomes: tracing back the history of photosynthetic life on Earth. *The Plant journal : for cell and molecular biology* **66**, 45-57 (2011).

42. R. I. Ponce-Toledo *et al.*, An Early-Branching Freshwater Cyanobacterium at the Origin of Plastids. *Current biology : CB* **27**, 386-391 (2017).

43. D. C. Price *et al.*, *Cyanophora paradoxa* Genome Elucidates Origin of Photosynthesis in Algae and Plants. *Science* **335**, 843-847 (2012).

44. J. M. Archibald, Genomic perspectives on the birth and spread of plastids. *Proc Natl Acad Sci U S A* **112**, 10147-10153 (2015).

45. J. R. Brown, Ancient horizontal gene transfer. *Nature reviews. Genetics* **4**, 121-132 (2003).

46. E. Kim, J. M. Archibald, Diversity and Evolution of Plastids and Their Genomes. *Plant Cell Monogr* **13**, 1-39 (2009).

47. N. Rolland *et al.*, The biosynthetic capacities of the plastids and integration between cytoplasmic and chloroplast processes. *Annual review of genetics* **46**, 233-264 (2012).

48. C. Shi *et al.*, Full transcription of the chloroplast genome in photosynthetic eukaryotes. *Scientific reports* **6**, 30135 (2016).

49. B. Llorente *et al.*, Selective pressure against horizontally acquired prokaryotic genes as a driving force of plastid evolution. *Scientific reports* **6**, 19036 (2016).

50. P. Mehrshahi, C. Johnny, D. DellaPenna, Redefining the metabolic continuity of chloroplasts and ER. *Trends in plant science* **19**, 501-507 (2014).

51. N. Tiller, R. Bock, The translational apparatus of plastids and its role in plant development. *Molecular plant* **7**, 1105-1120 (2014).

52. L. Sun *et al.*, Chloroplast Phylogenomic Inference of Green Algae Relationships. *Scientific reports* **6**, 20528 (2016).

53. C. Lemieux, C. Otis, M. Turmel, Chloroplast phylogenomic analysis resolves deep-level relationships within the green algal class Trebouxiophyceae. *BMC Evolutionary Biology* **14**, (2014).

54. F. Leliaert *et al.*, Chloroplast phylogenomic analyses reveal the deepest-branching lineage of the Chlorophyta, Palmophyllophyceae class. nov. *Scientific reports* **6**, 25367 (2016).

55. L. Fang, F. Leliaert, Z.-H. Zhang, D. Penny, B.-J. Zhong, Evolution of the Chlorophyta: Insights from chloroplast phylogenomic analyses. *Journal of Systematics and Evolution* **55**, 322-332 (2017).

56. M. Turmel, C. Otis, C. Lemieux, Dynamic Evolution of the Chloroplast Genome in the Green Algal Classes Pedinophyceae and Trebouxiophyceae. *Genome biology and evolution* **7**, 2062-2082 (2015).

57. J. L. Bowman, S. K. Floyd, K. Sakakibara, Green genes-comparative genomics of the green branch of life. *Cell* **129**, 229-234 (2007).

58. K. M. Kim, J. H. Park, D. Bhattacharya, H. S. Yoon, Applications of next-generation sequencing to unravelling the evolutionary history of algae. *Int J Syst Evol Microbiol* **64**, 333-345 (2014).

59. J. A. Raven, M. Giordano, J. Beardall, S. C. Maberly, Algal evolution in relation to atmospheric CO2: carboxylases, carbon-concentrating mechanisms and carbon oxidation cycles. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* **367**, 493-507 (2012).

60. S. Aubry, S. Kelly, B. M. Kumpers, R. D. Smith-Unna, J. M. Hibberd, Deep evolutionary comparison of gene expression identifies parallel recruitment of trans-factors in two independent origins of C4 photosynthesis. *PLoS genetics* **10**, e1004365 (2014).

61. R. Ming *et al.*, The pineapple genome and the evolution of CAM photosynthesis. *Nature genetics* **47**, 1435-1442 (2015).

62. F. Leliaert *et al.*, Phylogeny and Molecular Evolution of the Green Algae. *Critical Reviews in Plant Sciences* **31**, 1-46 (2012).

63. B. Becker, B. Marin, Streptophyte algae and the origin of embryophytes. *Annals of botany* **103**, 999-1004 (2009).

64. F. Leliaert, H. Verbruggen, F. W. Zechman, Into the deep: new discoveries at the base of the green plant phylogeny. *BioEssays : news and reviews in molecular, cellular and developmental biology* **33**, 683-692 (2011).

65. B. Zhong, L. Sun, D. Penny, The Origin of Land Plants: A Phylogenomic Perspective. *Evol Bioinform Online* **11**, 137-141 (2015).

66. B. J. Olson, A. M. Nedelcu, Co-option during the evolution of multicellular and developmental complexity in the volvocine green algae. *Current opinion in genetics & development* **39**, 107-115 (2016).

67. E. R. Hanschen *et al.*, The Gonium pectorale genome demonstrates co-option of cell cycle regulation during the evolution of multicellularity. *Nature communications* **7**, 11370 (2016).

68. A. de Mendoza *et al.*, Transcription factor evolution in eukaryotes and the assembly of the regulatory toolkit in multicellular lineages. *Proc Natl Acad Sci U S A* **110**, E4858-4866 (2013).

69. S. Thiriet-Rupert *et al.*, Transcription factors in microalgae: genome-wide prediction and comparative analysis. *BMC Genomics* **17**, 282 (2016).

70. S. Geng, P. De Hoff, J. Umen, Evolution of Sexes from an Ancestral Mating-Type Specification Pathway. *PLoS Biol* **12**, (2014).

71. H. Moreau *et al.*, Gene functionalities and genome structure in Bathycoccus prasinos reflect cellular specializations at the base of the green lineage. *Genome Biol* **13**, R74 (2012).

72. M. J. van Baren *et al.*, Evidence-based green algal genomics reveals marine diversity and ancestral characteristics of land plants. *BMC Genomics* **17**, 267 (2016).

73. M. Iwai, M. Yokono, Light-harvesting antenna complexes in the moss Physcomitrella patens: implications for the evolutionary transition from green algae to land plants. *Current opinion in plant biology* **37**, 94-101 (2017).

74. J. L. Bowman *et al.*, Insights into Land Plant Evolution Garnered from the Marchantia polymorpha Genome. *Cell* **171**, 287-304 e215 (2017).

75. J. A. Banks *et al.*, The Selaginella Genome Identifies Genetic Changes Associated with the Evolution of Vascular Plants. *Science* **332**, 960-963 (2011).

76. P. M. Delaux *et al.*, Algal ancestor of land plants was preadapted for symbiosis. *Proc Natl Acad Sci U S A* **112**, 13390-13395 (2015).

77. J. Salse, Ancestors of modern plant crops. *Current opinion in plant biology* **30**, 134-142 (2016).

78. F. Murat, A. Armero, C. Pont, C. Klopp, J. Salse, Reconstructing the genome of the most recent common ancestor of flowering plants. *Nature genetics* **49**, 490-496 (2017).

79. S. A. Rensing, Gene duplication as a driver of plant morphogenetic evolution. *Current opinion in plant biology* **17**, 43-48 (2014).

80. L. M. Evans *et al.*, Population genomics of Populus trichocarpa identifies signatures of selection and adaptive trait associations. *Nature genetics*, (2014).

81. D. Zhao, A. A. Ferguson, N. Jiang, What makes up plant genomes: The vanishing line between transposable elements and genes. *Biochimica et biophysica acta* **1859**, 366-380 (2016).

82. J. L. Bennetzen, H. Wang, The contributions of transposable elements to the structure, function, and evolution of plant genomes. *Annual review of plant biology* **65**, 505-530 (2014).

83. X. Cui, X. Cao, Epigenetic regulation and functional exaptation of transposable elements in higher plants. *Current opinion in plant biology* **21C**, 83-88 (2014).

84. D. Lisch, How important are transposons for plant evolution? *Nature reviews. Genetics* **14**, 49-61 (2013).

85. N. V. Fedoroff, Transposable Elements, Epigenetics, and Genome Evolution. *Science* **338**, 758-767 (2012).

86. B. Nystedt *et al.*, The Norway spruce genome sequence and conifer genome evolution. *Nature* **497**, 579-584 (2013).

87. Z. Zhao *et al.*, Genome-wide analysis of tandem repeats in plants and green algae. *G3 (Bethesda)* **4**, 67-78 (2014).

88. R. Chodat, Sur la structure et la biologie de deux algues pelagiques. *Journal de Botanique* **10**, 341-349 (1896).

89. K. B. Blackburn, *Botryococcus* and the Algal Coals Part I: A Reinvestigation of the Alga *Botryococcus braunii* Kutzing. *Trans Roy Soc Edin* **58**, 841-854 (1936).

90. S. P. Chu, The Influence of the Mineral Composition of the Medium on the Growth of Planktonic Algae: Part I. Methods and Culture Media. *Journal of Ecology* **30**, 284-325 (1942).

91. J. H. Belcher, G. E. Fogg, Biochemical evidence of the affinities of *Botryococcus*. *The New phytologist* **54**, 81-83 (1955).

92. A. Traverse, Occurrence of the Oil-Forming Alga *Botryococcus* in Lignites and Other Tertiary Sediments. *Micropaleontology* **1**, 343-349 (1955).

93. E. Gelpi, J. Oro, H. J. Schneider, E. O. Bennett, Olefins of High Molecular Weight in Two Microscopic Algae. *Science* **161**, 700-701 (1968).

94. J. R. Maxwell, A. G. Douglas, G. Eglinton, A. McCormick, The botryococcenes – hydrocarbons of novel structure from the alga *Botryococcus braunii*, Kutzing. *Phytochemistry* **7**, 2157-2171 (1968).

95. A. C. Brown, B. A. Knights, E. Conway, Hydrocarbon content and its relationship to physiological state in the green alga *Botryococcus braunii*. *Phytochemistry* **8**, 543-547 (1969).

96. J. Murray, A. Thomson, Hydrocarbon production in *Anacystis montana* and *Botryococcus braunii*. *Phytochemistry* **16**, 465-468 (1977).

97. C. Largeau, E. Casadevall, C. Berkaloff, P. Dhamelincourt, Sites of accumulation and composition of hydrocarbons in *Botryococcus braunii*. *Phytochemistry* **19**, 1043-1051 (1980).

98. C. Largeau, E. Casadevall, C. Berkaloff, The biosynthesis of long-chain hydrocarbons in the green alga *Botryococcus braunii*. *Phytochemistry* **19**, 1081-1085 (1980).

99. L. V. Wake, L. W. Hillen, Study of a "Bloom" of the Oil-Rich Alga *Botryococcus braunii* in the Darwin River Reservoir. *Biotechnology and Bioengineering* **22**, 1637-1657 (1980).

100. L. W. Hillen, G. Pollard, L. V. Wake, N. White, Hydrocracking of the Oils of *Botryococcus braunii* to Transport Fuels. *Biotechnology and Bioengineering* **24**, 193-205 (1982).

101. F. R. Wolf, *Botryococcus braunii* An Unusual Hydrocarbon-Producing Alga. *Appl Biochem Biotechnol* **8**, 249-260 (1983).

102. C. Berkaloff, B. Rousseau, Variability of cell wall structure and hydrocarbon type in different strains of *Botryococcus braunii*. *Journal of Phycology* **20**, 377-389 (1984).

103. P. Metzger, C. Berkaloff, E. Casadevall, A. Coute, Alkadiene- and botryococcene-producing races of wild strains of *Botryococcus braunii*. *Phytochemistry* **24**, 2305-2312 (1985).

104. P. Metzger, E. Casadevall, Lycopadiene, a tetraterpenoid hydrocarbon from new strains of the green alga *Botryococcus braunii*. *Tetrahedron Letters* **28**, 3931-3934 (1987).

105. M. Glikson, K. Lindsay, J. Saxby, *Botryococcus* - A planktonic green alga, the source of petroleum through the ages: Transmission electron microscopical studies of oil shales and petroleum source rocks. *Organic Geochemistry* **14**, 595-608 (1989).

106. D. Guy-Ohlson, *Botryococcus* as an aid in the interpretation of palaeoenvironment and depositional processes. *Review of Palaebotany and Palynology* **71**, 1-15 (1992).

107. S. Sawayama, S. Inoue, S. Yokoyama, Phylogenetic position of *Botryococcus braunii* (Chlorophyceae) based on small subunit ribosomal RNA sequence data. *J Phycol* **31**, 419-420 (1995).

108. G. W. Beakes, A. L. Cleary, Visualization of plastids and lipophilic components in living colonies of a wild strain of the hydrocarbon-forming green alga Botryococcus by laser scanning confocal microscopy. *Journal of Applied Phycology* **10**, 435-446 (1998).

109. J. Vioque, T. Sirakova, P. E. Kolattukudy, The malate dehydrogenase gene from *Botryococcus braunii* (Chlorophyta, Chlorophyceae): Cloning, sequence analysis, and expression in *Escherichia coli*. *Journal of Phycology* **35**, 121-127 (1999).

110. S. Okada, T. P. Devarenne, J. Chappell, Molecular characterization of squalene synthase from the green microalga Botryococcus braunii, race B. *Arch Biochem Biophys* **373**, 307-317 (2000).

111. A. Banerjee, R. Sharma, Y. Chisti, U. C. Banerjee, *Botryococcus braunii*: A Renewable Source of Hydrocarbons and Other Chemicals. *Critical Reviews in Biotechnology* **22**, 245-279 (2002).

112. Y. Sato, Y. Ito, S. Okada, M. Murakami, H. Abe, Biosynthesis of the triterpenoids, botryococcenes and tetramethylsqualene in the B race of Botryococcus braunii via the non-mevalonate pathway. *Tetrahedron Letters* **44**, 7035-7037 (2003).

113. S. Okada, T. P. Devarenne, M. Murakami, H. Abe, J. Chappell, Characterization of botryococcene synthase enzyme activity, a squalene synthase-like activity from the green

microalga Botryococcus braunii, Race B. *Archives of Biochemistry and Biophysics* **422**, 110-118 (2004).

114. H. H. Senousy, G. W. Beakes, E. Hack, Phylogenetic Placement of Botryococcus Braunii (Trebouxiophyceae) and Botryococcus Sudeticus Isolate Utex 2629 (Chlorophyceae)1. *Journal of Phycology* **40**, 412-423 (2004).

115. P. Metzger, C. Largeau, Botryococcus braunii: a rich source for hydrocarbons and related ether lipids. *Appl Microbiol Biotechnol* **66**, 486-496 (2005).

116. R. de Mesmay, P. Metzger, V. Grossi, S. Derenne, Mono- and dicyclic unsaturated triterpenoid hydrocarbons in sediments from Lake Masoko (Tanzania) widely extend the botryococcene family. *Organic Geochemistry* **39**, 879-893 (2008).

117. P. Metzger, M. N. Rager, C. Fosse, Braunicetals: acetals from condensation of macrocyclic aldehydes and terpene diols in Botryococcus braunii. *Phytochemistry* **69**, 2380-2386 (2008).

118. T. L. Weiss *et al.*, Phylogenetic Placement, Genome Size, and Gc Content of the Liquid-Hydrocarbon-Producing Green Microalga Botryococcus Braunii Strain Berkeley (Showa) (Chlorophyta). *Journal of Phycology* **46**, 534-540 (2010).

119. T. L. Weiss, J. S. Johnston, K. Fujisawa, S. Okada, T. P. Devarenne, Genome size and phylogenetic analysis of the A and L races of Botryococcus braunii. *Journal of Applied Phycology* **23**, 833-839 (2011).

120. T. D. Niehaus *et al.*, Identification of unique mechanisms for triterpene biosynthesis in *Botryococcus braunii*. *PNAS* **108**, 12260-12265 (2011).

121. T. D. Niehaus *et al.*, Functional identification of triterpene methyltransferases from Botryococcus braunii race B. *The Journal of biological chemistry* **287**, 8163-8173 (2012).

122. I. Molnar *et al.*, Bio-crude transcriptomics: Gene discovery and metabolic network reconstruction for the biosynthesis of the terpenome of the hydrocarbon oil-producing green alga, *Botryococcus braunii* race B (Showa). *BMC Genomics* **13**, (2012).

123. M. Baba, M. Ioki, N. Nakajima, Y. Shiraiwa, M. M. Watanabe, Transcriptome analysis of an oil-rich race A strain of Botryococcus braunii (BOT-88-2) by de novo assembly of pyrosequencing cDNA reads. *Bioresource technology* **109**, 282-286 (2012).

124. M. Ioki, M. Baba, N. Nakajima, Y. Shiraiwa, M. M. Watanabe, Transcriptome analysis of an oil-rich race B strain of Botryococcus braunii (BOT-22) by de novo assembly of pyrosequencing cDNA reads. *Bioresource technology* **109**, 292-296 (2012).

125. M. Ioki *et al.*, Modes of hydrocarbon oil biosynthesis revealed by comparative gene expression analysis for race A and race B strains of Botryococcus braunii. *Bioresource technology* **109**, 271-276 (2012).

126. M. Kawachi, T. Tanoi, M. Demura, K. Kaya, M. M. Watanabe, Relationship between hydrocarbons and molecular phylogeny of Botryococcus braunii. *Algal Research* **1**, 114-119 (2012).

127. T. L. Weiss *et al.*, Colony Organization in the Green Alga Botryococcus braunii (Race B) Is Specified by a Complex Extracellular Matrix. *Eukaryotic cell* **11**, 1424-1440 (2012).

128. R. Suzuki *et al.*, Transformation of Lipid Bodies Related to Hydrocarbon Accumulation in a Green Alga, Botryococcus braunii (Race B). *PLoS ONE* **8**, e81626 (2013).

129. M. Hirose, F. Mukaida, S. Okada, T. Noguchi, Active Hydrocarbon Biosynthesis and Accumulation in a Green Alga, Botryococcus braunii (Race A). *Eukaryotic cell* **12**, 1132-1141 (2013).

130. H. S. Kim, T. L. Weiss, H. R. Thapa, T. P. Devarenne, A. Han, A microfluidic photobioreactor array demonstrating high-throughput screening for microalgal oil production. *Lab on a Chip*, (2014).

131. J. K. Volkman, Acyclic isoprenoid biomarkers and evolution of biosynthetic pathways in green microalgae of the genus Botryococcus. *Organic Geochemistry* **75**, 36-47 (2014).

132. H. Berrios, M. Zapata, M. Rivas, A method for genetic transformation of Botryococcus braunii using a cellulase pretreatment. *Journal of Applied Phycology*, (2015).

133. Y. Tanabe *et al.*, A novel alphaproteobacterial ectosymbiont promotes the growth of the hydrocarbon-rich green alga Botryococcus braunii. *Scientific reports* **5**, 10467 (2015).

134. K. J. Jones *et al.*, Draft Genome Sequences of Achromobacter piechaudii GCS2, Agrobacterium sp. Strain SUL3, Microbacterium sp. Strain GCS4, Shinella sp. Strain GWS1, and Shinella sp. Strain SUS2 Isolated from Consortium with the Hydrocarbon-Producing Alga Botryococcus braunii. *Genome Announc* **4**, (2016).

135. H. R. Thapa *et al.*, A squalene synthase-like enzyme initiates production of tetraterpenoid hydrocarbons in Botryococcus braunii Race L. *Nature communications* **7**, 11198 (2016).

136. O. Blifernez-Klassen *et al.*, Complete Chloroplast and Mitochondrial Genome Sequences of the Hydrocarbon Oil-Producing Green Microalga Botryococcus braunii Race B (Showa). *Genome Announc* **4**, e00524-00516 (2016).

137. D. R. Browne *et al.*, Draft Nuclear Genome Sequence of the Liquid Hydrocarbon-Accumulating Green Microalga Botryococcus braunii Race B (Showa). *Genome Announc* **5**, (2017).

138. C. Sambles *et al.*, Metagenomic analysis of the complex microbial consortium associated with cultures of the oil-rich alga Botryococcus braunii. *Microbiologyopen* **6**, (2017).

139. X. Y. Deng *et al.*, Identification and analysis of microRNAs in Botryococcus braunii using high-throughput sequencing. *Aquatic Biology* **26**, 41-48 (2017).

140. R. Suzuki, I. Nishii, S. Okada, T. Noguchi, 3D reconstruction of endoplasmic reticulum in a hydrocarbon-secreting green alga, Botryococcus braunii (Race B). *Planta*, (2017).

141. T. E. van den Berg, B. van Oort, R. Croce, Light-harvesting complexes of Botryococcus braunii. *Photosynth Res*, (2017).

142. M. Tatli *et al.*, Polysaccharide associated protein (PSAP) from the green microalga Botryococcus braunii is a unique extracellular matrix hydroxyproline-rich glycoprotein. *Algal Research* **29**, 92-103 (2018).

143. M. Calvin, Fuel oils from euphorbs and other plants. *Botanical Journal of the Linnean Society* **94**, 97-110 (1987).

144. L. Meher, D. Vidyasagar, S. Naik, Technical aspects of biodiesel production by transesterification—a review. *Renewable and Sustainable Energy Reviews* **10**, 248-268 (2006).

145. G. Frazzetto, White biotechnology. *EMBO reports* **4**, 835-837 (2003).

146. A. J. Ragauskas *et al.*, The path forward for biofuels and biomaterials. *Science* **311**, 484-489 (2006).

147. A. W. Larkum, Limitations and prospects of natural photosynthesis for bioenergy production. *Current opinion in biotechnology* **21**, 271-276 (2010).

148. X. G. Zhu, S. P. Long, D. R. Ort, Improving photosynthetic efficiency for greater yield. *Annual review of plant biology* **61**, 235-261 (2010).

149. R. H. Wijffels, M. J. Barbosa, An outlook on microalgal biofuels. *Science* **329**, 796-799 (2010).

150. Y. Chisti, Constraints to commercialization of algal fuels. *J Biotechnol* **167**, 201-214 (2013).

151. M. Yoshida, Y. Tanabe, N. Yonezawa, M. M. Watanabe, Energy innovation potential of oleaginous microalgae. *Biofuels* **3**, 761-781 (2012).

152. N. Pragya, K. K. Pandey, P. K. Sahoo, A review on harvesting, oil extraction and biofuels production technologies from microalgae. *Renewable and Sustainable Energy Reviews* **24**, 159-171 (2013).

153. I. Rawat, R. Ranjith Kumar, T. Mutanda, F. Bux, Biodiesel from microalgae: A critical evaluation from laboratory to large scale production. *Applied Energy* **103**, 444-467 (2013).

154. R. Schurr, A. Kuehnle, Microalgae Crop Improvement: Tools for Quality Control and Molecular Breeding. *Industrial Biotechnology* **10**, 237-243 (2014).

155. A. Miara, P. T. Pienkos, M. Bazilian, R. Davis, J. Macknick, Planning for Algal Systems: An Energy-Water-Food Nexus Perspective. *Industrial Biotechnology* **10**, 202-211 (2014).

156. J. W. Moody, C. M. McGinty, J. C. Quinn, Global evaluation of biofuel potential from microalgae. *PNAS* **111**, 8691-8696 (2014).

157. E. R. Venteris, R. L. Skaggs, M. S. Wigmosta, A. M. Coleman, Regional algal biofuel production potential in the coterminous United States as affected by resource availability trade-offs. *Algal Research* **5**, 215-225 (2014).

158. B. Wang, C. Q. Lan, M. Horsman, Closed photobioreactors for production of microalgal biomasses. *Biotechnology advances* **30**, 904-912 (2012).

159. F. G. Acien, J. M. Fernandez, J. J. Magan, E. Molina, Production cost of a real microalgae production plant and strategies to reduce it. *Biotechnology advances* **30**, 1344-1353 (2012).

160. J. W. Richardson, M. D. Johnson, J. L. Outlaw, Economic comparison of open pond raceways to photo bio-reactors for profitable production of algae for transportation fuels in the Southwest. *Algal Research* **1**, 93-100 (2012).

161. D. Klein-Marcuschamer *et al.*, Technoeconomic analysis of renewable aviation fuel from microalgae,Pongamia pinnata, and sugarcane. *Biofuels, Bioproducts and Biorefining* **7**, 416-428 (2013).

162. J. M. Abodeely *et al.*, Assessment of algal farm designs using a dynamic modular approach. *Algal Research* **5**, 264-273 (2014).

163. C. G. Gutiérrez-Arriaga, M. Serna-González, J. M. Ponce-Ortega, M. M. El-Halwagi, Sustainable Integration of Algal Biodiesel Production with Steam Electric Power Plants for Greenhouse Gas Mitigation. *ACS Sustainable Chemistry & Engineering* **2**, 1388-1403 (2014).

164. Y. Zhu, K. O. Albrecht, D. C. Elliott, R. T. Hallen, S. B. Jones, Development of hydrothermal liquefaction and upgrading technologies for lipid-extracted algae conversion to liquid fuels. *Algal Research* **2**, 455-464 (2013).

165. C. Silva *et al.*, Commercial-Scale Biodiesel Production from Algae. *Industrial & Engineering Chemistry Research* **53**, 5311-5324 (2014).

166. X. Liu *et al.*, Pilot-scale data provide enhanced estimates of the life cycle energy and emissions profile of algae biofuels produced via hydrothermal liquefaction. *Bioresource technology* **148C**, 163-171 (2013).

167. R. M. Handler, D. R. Shonnard, T. N. Kalnes, F. S. Lupton, Life cycle assessment of algal biofuels: Influence of feedstock cultivation systems and conversion platforms. *Algal Research* **4**, 105-115 (2014).

168. N. D. Orfield, R. B. Levine, G. A. Keoleian, S. A. Miller, P. E. Savage, Growing Algae for Biodiesel on Direct Sunlight or Sugars: A Comparative Life Cycle Assessment. *ACS Sustainable Chemistry & Engineering* **3**, 386-395 (2015).

169. P. Kandimalla, S. Desi, H. Vurimindi, Mixotrophic cultivation of microalgae using industrial flue gases for biodiesel production. *Environ Sci Pollut Res Int* **23**, 9345-9354 (2016).

170. R. Honda, J. Boonnorat, C. Chiemchaisri, W. Chiemchaisri, K. Yamamoto, Carbon dioxide capture and nutrients removal utilizing treated sewage by concentrated microalgae cultivation in a membrane photobioreactor. *Bioresource technology* **125**, 59-64 (2012).

171. O. Fenton, D. Ó hUallacháin, Agricultural nutrient surpluses as potential input sources to grow third generation biomass (microalgae): A review. *Algal Research* **1**, 49-56 (2012).

172. P. Biller *et al.*, Nutrient recycling of aqueous phase for microalgae cultivation from the hydrothermal liquefaction process. *Algal Research* **1**, 70-76 (2012).

173. M. T. Guarnieri, P. T. Pienkos, Algal omics: unlocking bioproduct diversity in algae cell factories. *Photosynth Res* **123**, 255-263 (2015).

174. S. Leu, S. Boussiba, Advances in the Production of High-Value Products by Microalgae. *Industrial Biotechnology* **10**, 169-183 (2014).

175. T. Dong *et al.*, Combined algal processing: A novel integrated biorefinery process to produce algal biofuels and bioproducts. *Algal Research*, (2016).

176. A. de la Escosura, C. Briones, K. Ruiz-Mirazo, The systems perspective at the crossroads between chemistry and biology. *J Theor Biol* **381**, 11-22 (2015).

177. T. J. Erb, P. R. Jones, A. Bar-Even, Synthetic metabolism: metabolic engineering meets enzyme design. *Current opinion in chemical biology* **37**, 56-62 (2017).

178. V. Chubukov, A. Mukhopadhyay, C. J. Petzold, J. D. Keasling, H. G. Martín, Synthetic and systems biology for microbial production of commodity chemicals. *npj Systems Biology and Applications* **2**, 16009 (2016).

179. B. Lowry, C. T. Walsh, C. Khosla, Reconstitution of Metabolic Pathways: Insights into Nature's Chemical Logic. *Synlett* **26**, 1008-1025 (2015).

180. M. E. Guazzaroni, R. Silva-Rocha, R. J. Ward, Synthetic biology approaches to improve biocatalyst identification in metagenomic library screening. *Microb Biotechnol* **8**, 52-64 (2015).

181. E. T. Wurtzel, T. M. Kutchan, Plant metabolism, the diverse chemistry set of the future. *Science* **353**, 1232-1236 (2016).

182. L. M. Dersch, V. Beckers, C. Wittmann, Green pathways: Metabolic network analysis of plant systems. *Metab Eng* **34**, 1-24 (2016).

183. T. J. Erb, J. Zarzycki, Biochemical and synthetic biology approaches to improve photosynthetic CO2-fixation. *Current opinion in chemical biology* **34**, 72-79 (2016).

184. H. Shi, J. Schwender, Mathematical models of plant metabolism. *Current opinion in biotechnology* **37**, 143-152 (2016).

185. S. Imam *et al.*, A refined genome-scale reconstruction of Chlamydomonas metabolism provides a platform for systems-level analyses. *The Plant journal : for cell and molecular biology* **84**, 1239-1256 (2015).

186. S. P. Chapman, C. M. Paget, G. N. Johnson, J. M. Schwartz, Flux balance analysis reveals acetate metabolism modulates cyclic electron flow and alternative glycolytic pathways in Chlamydomonas reinhardtii. *Frontiers in plant science* **6**, 474 (2015).

187. N. Loira *et al.*, Reconstruction of the microalga Nannochloropsis salina genome-scale metabolic model with applications to lipid production. *BMC Syst Biol* **11**, 66 (2017).

188. J. Levering *et al.*, Genome-Scale Model Reveals Metabolic Basis of Biomass Partitioning in a Model Diatom. *PLoS One* **11**, e0155038 (2016).

189. J. H. Wisecaver *et al.*, A Global Coexpression Network Approach for Connecting Genes to Specialized Metabolic Pathways in Plants. *The Plant cell* **29**, 944-959 (2017).

190. M. A. Scaife, A. G. Smith, Towards developing algal synthetic biology. *Biochem Soc Trans* **44**, 716-722 (2016).

191. S. E. Shin *et al.*, CRISPR/Cas9-induced knockout and knock-in mutations in Chlamydomonas reinhardtii. *Scientific reports* **6**, 27810 (2016).

192. P. M. Shih *et al.*, A robust gene-stacking method utilizing yeast assembly for plant synthetic biology. *Nature communications* **7**, 13215 (2016).

193. Y. Liang *et al.*, Endoribonuclease-Based Two-Component Repressor Systems for Tight Gene Expression Control in Plants. *ACS Synth Biol*, (2017).

194. X. Tang, J. Lee, W. N. Chen, Engineering the fatty acid metabolic pathway in Saccharomyces cerevisiae for advanced biofuel production. *Metabolic Engineering Communications* **2**, 58-66 (2015).

195. W. Runguphan, J. D. Keasling, Metabolic engineering of Saccharomyces cerevisiae for production of fatty acid-derived biofuels and chemicals. *Metab Eng* **21**, 103-113 (2014).

196. A. Reider Apel *et al.*, A Cas9-based toolkit to program gene expression in Saccharomyces cerevisiae. *Nucleic acids research*, (2016).

197. Andrew A. Horwitz *et al.*, Efficient Multiplexed Integration of Synergistic Alleles and Metabolic Pathways in Yeasts via CRISPR-Cas. *Cell Systems*, (2015).

198. B. Chen, D. Y. Lee, M. W. Chang, Combinatorial metabolic engineering of Saccharomyces cerevisiae for terminal alkene production. *Metab Eng* **31**, 53-61 (2015).

199. A. L. Meadows *et al.*, Rewriting yeast central carbon metabolism for industrial isoprenoid production. *Nature* **537**, 694-697 (2016).

200. C. M. Schwartz, M. S. Hussain, M. Blenner, I. Wheeldon, Synthetic RNA Polymerase III Promoters Facilitate High-Efficiency CRISPR-Cas9-Mediated Genome Editing in Yarrowia lipolytica. *ACS Synth Biol*, (2016).

201. L. Liu, A. Pan, C. Spofford, N. Zhou, H. S. Alper, An evolutionary metabolic engineering approach for enhancing lipogenesis in Yarrowia lipolytica. *Metab Eng* **29**, 36-45 (2015).

202. K. Qiao *et al.*, Engineering lipid overproduction in the oleaginous yeast Yarrowia lipolytica. *Metab Eng* **29**, 56-65 (2015).

203. A. A. Green *et al.*, Complex cellular logic computation using ribocomputing devices. *Nature* **548**, 117-121 (2017).

204. J. G. Zalatan *et al.*, Engineering Complex Synthetic Transcriptional Programs with CRISPR RNA Scaffolds. *Cell* **160**, 339-350 (2015).

205. R. W. Bradley, M. Buck, B. Wang, Recognizing and engineering digital-like logic gates and switches in gene regulatory networks. *Curr Opin Microbiol* **33**, 74-82 (2016).

206. J. Shendure *et al.*, DNA sequencing at 40: past, present and future. *Nature* **550**, 345-353 (2017).

207. S. Goodwin, J. D. McPherson, W. R. McCombie, Coming of age: ten years of next-generation sequencing technologies. *Nature reviews. Genetics* **17**, 333-351 (2016).

208. M. L. Metzker, Sequencing technologies - the next generation. *Nature reviews. Genetics* **11**, 31-46 (2010).

209. M. Pop, S. L. Salzberg, Bioinformatics challenges of new sequencing technology. *Trends in genetics : TIG* **24**, 142-149 (2008).

210. M. G. Ross *et al.*, Characterizing and measuring bias in sequence data. *Genome Biology* **14**, (2013).

211. M. O. Carneiro *et al.*, Pacific biosciences sequencing technology for genotyping and variation discovery in human data. *BMC Genomics* **13**, 375 (2012).

212. Z. Li *et al.*, Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph. *Briefings in functional genomics* **11**, 25-37 (2012).

213. S. Batzoglou *et al.*, ARACHNE: A Whole-Genome Shotgun Assembler. *Genome research* **12**, 177-189 (2002).

214. P. A. Pevzner, H. Tang, M. S. Waterman, An Eulerian path approach to DNA fragment assembly. *Proc Natl Acad Sci U S A* **98**, 9748-9753 (2001).

215. J. R. Miller, S. Koren, G. Sutton, Assembly algorithms for next-generation sequencing data. *Genomics* **95**, 315-327 (2010).

216. D. Earl *et al.*, Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome research* **21**, 2224-2241 (2011).

217. G. Narzisi, B. Mishra, Comparing de novo genome assembly: the long and short of it. *PLoS One* **6**, e19175 (2011).

218. F. A. Simao, R. M. Waterhouse, P. Ioannidis, E. V. Kriventseva, E. M. Zdobnov, BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, (2015).

219. J. T. Simpson, R. Durbin, Efficient de novo assembly of large genomes using compressed data structures. *Genome research* **22**, 549-556 (2012).

220. A. Bankevich *et al.*, SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J Comput Biol* **19**, 455-477 (2012).

221. A. V. Zimin *et al.*, The MaSuRCA genome assembler. *Bioinformatics* **29**, 2669-2677 (2013).

222. N. I. Weisenfeld *et al.*, Comprehensive variation discovery in single human genomes. *Nature genetics* **46**, 1350-1355 (2014).

223. R. Chikhi, A. Limasset, P. Medvedev, Compacting de Bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics* **32**, i201-i208 (2016).

224. S. D. Jackman *et al.*, ABySS 2.0: resource-efficient assembly of large genomes using a Bloom filter. *Genome research* **27**, 768-777 (2017).

225. C. S. Chin *et al.*, Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nature methods* **10**, 563-569 (2013).

226. G. M. Kamath, I. Shomorony, F. Xia, T. A. Courtade, D. N. Tse, HINGE: long-read assembly achieves optimal repeat resolution. *Genome research* **27**, 747-756 (2017).

227. C. S. Chin *et al.*, Phased diploid genome assembly with single-molecule real-time sequencing. *Nature methods* **13**, 1050-1054 (2016).

228. S. Koren *et al.*, Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome research* **27**, 722-736 (2017).

229. Y. Lin *et al.*, Assembly of long error-prone reads using de Bruijn graphs. *Proc Natl Acad Sci U S A* **113**, E8396-E8405 (2016).

230. M. Kolmogorov, J. Yuan, Y. Lin, P. Pevzner, Assembly of Long Error-Prone Reads Using Repeat Graphs. *bioRxiv*, (2018).

231. V. Deshpande, E. D. Fung, S. Pham, V. Bafna, Cerulean: A hybrid assembly using high throughput short and long reads. *arXiv* **1307.7933v1**, (2013).

232. C. Ye, C. M. Hill, S. Wu, J. Ruan, Z. S. Ma, DBG2OLC: Efficient Assembly of Large Genomes Using Long Erroneous Reads of the Third Generation Sequencing Technologies. *Scientific reports* **6**, 31900 (2016).

233. A. V. Zimin *et al.*, Hybrid assembly of the large and highly repetitive genome of Aegilops tauschii, a progenitor of bread wheat, with the MaSuRCA mega-reads algorithm. *Genome research* **27**, 787-792 (2017).

234. M. Scholz, C. C. Lo, P. S. Chain, Improved assemblies using a source-agnostic pipeline for MetaGenomic Assembly by Merging (MeGAMerge) of contigs. *Scientific reports* **4**, 6480 (2014).

235. A. H. Wences, M. C. Schatz, Metassembler: merging and optimizing de novo genome assemblies. *Genome Biol* **16**, 207 (2015).

236. H. Alhakami, H. Mirebrahim, S. Lonardi, A comparative evaluation of genome assembly reconciliation tools. *Genome Biol* **18**, 93 (2017).

237. H. Li, N. Homer, A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in bioinformatics* **11**, 473-483 (2010).

238. K. Reinert, B. Langmead, D. Weese, D. J. Evers, Alignment of Next-Generation Sequencing Reads. *Annu Rev Genomics Hum Genet* **16**, 133-151 (2015).

239. W. Li, J. Feudenberg, P. Miramontes, Diminishing return for increased Mappability with longer sequencing reads: implications of the k-mer distributions in the human genome. *BMC bioinformatics* **15**, 1-12 (2014).

240. M. Smolka, P. Rescheneder, M. C. Schatz, A. von Haeseler, F. J. Sedlazeck, Teaser: Individualized benchmarking and optimization of read mapping results for NGS data. *Genome Biol* **16**, 235 (2015).

241. M. Chaisson, G. Tesler, Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. *BMC bioinformatics* **13**, (2012).

242. D. Kim, B. Langmead, S. L. Salzberg, HISAT: a fast spliced aligner with low memory requirements. *Nature methods* **12**, 357-360 (2015).

243. B. Liu, H. Guo, M. Brudno, Y. Wang, deBGA: read alignment with de Bruijn graph-based seed and extension. *Bioinformatics* **32**, 3224-3232 (2016).

244. I. Sovic *et al.*, Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nature communications* **7**, 11307 (2016).

245. S. Deorowicz, A. Debudaj-Grabysz, A. Gudys, S. Grabowski, Whisper: Read sorting allows robust mapping of sequencing data. *bioRxiv*, (2017).

246. H. Li, Minimap2: versatile pairwise alignment for nucleotide sequences. *arXiv* **1708**, (2017).

247. N. Nagarajan *et al.*, Finishing genomes with limited resources: lessons from an ensemble of microbial genomes. *BMC Genomics* **11**, 242 (2010).

248. M. Hunt, C. Newbold, M. Berriman, T. D. Otto, A comprehensive evaluation of assembly scaffolding tools. *Genome Biol* **15**, R42 (2014).

249. A. C. English *et al.*, Mind the gap: upgrading genomes with Pacific Biosciences RS long-read sequencing technology. *PLoS One* **7**, e47768 (2012).

250. H. Lee *et al.*, Error correction and assembly complexity of single molecule sequencing reads. (2014).

251. K. Sahlin, F. Vezzi, B. Nystedt, J. Lundeberg, L. Arvestad, BESST - Efficient scaffolding of large fragmented assemblies. *BMC bioinformatics* **15**, (2014).

252. M. S. Roth *et al.*, Chromosome-level genome assembly and transcriptome of the green alga Chromochloris zofingiensis illuminates astaxanthin production. *Proc Natl Acad Sci U S A* **114**, E4296-E4305 (2017).

253. M. Boetzer, W. Pirovano, SSPACE-LongRead: scaffolding bacterial draft genomes using long read sequence information. *BMC bioinformatics* **15**, 211 (2014).

254. R. L. Warren *et al.*, LINKS: Scalable, alignment-free scaffolding of draft genomes with long reads. *Gigascience* **4**, 35 (2015).

255. M. Boetzer, W. Pirovano, Toward almost closed genomes with GapFiller. *Genome Biol* **13**, R56 (2012).

256. V. C. Piro *et al.*, FGAP: an automated gap closing tool. *BMC Research Notes* **7**, 1-5 (2014).

257. D. Paulino *et al.*, Sealer: a scalable gap-closing application for finishing draft genomes. *BMC bioinformatics* **16**, 230 (2015).

258. R. Ronen, C. Boucher, H. Chitsaz, P. Pevzner, SEQuel: improving the accuracy of genome assemblies. *Bioinformatics* **28**, i188-196 (2012).

259. B. J. Walker *et al.*, Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS One* **9**, e112963 (2014).

260. R. Vaser, I. Sovic, N. Nagarajan, M. Sikic, Fast and accurate de novo genome assembly from long uncorrected reads. *Genome research* **27**, 737-746 (2017).

261. P. E. Compeau, P. A. Pevzner, G. Tesler, How to apply de Bruijn graphs to genome assembly. *Nat Biotechnol* **29**, 987-991 (2011).

262. J. T. Simpson *et al.*, ABySS: a parallel assembler for short read sequence data. *Genome research* **19**, 1117-1123 (2009).

263. J. E. Gallo, J. F. Munoz, E. Misas, J. G. McEwen, O. K. Clay, The complex task of choosing a de novo assembly: lessons from fungal genomes. *Comput Biol Chem* **53 Pt A**, 97-107 (2014).

264. L. Fu, B. Niu, Z. Zhu, S. Wu, W. Li, CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150-3152 (2012).

265. R. R. Wick, M. B. Schultz, J. Zobel, K. E. Holt, Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics* **31**, 3350-3352 (2015).

266. P. Jaccard, The distribution of the flora in the alpine zone. *The New phytologist* **11**, 37-50 (1912).

267. D. Mapleson, G. Garcia Accinelli, G. Kettleborough, J. Wright, B. J. Clavijo, KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies. *Bioinformatics* **33**, 574-576 (2017).

268. M. Hunt *et al.*, REAPR: a universal tool for genome assembly evaluation. *Genome Biol* **14**, R47 (2013).

269. M. Yandell, D. Ence, A beginner's guide to eukaryotic genome annotation. *Nature reviews. Genetics* **13**, 329-342 (2012).

270. E. L. Sonnhammer, G. Ostlund, InParanoid 8: orthology analysis between 273 proteomes, mostly eukaryotic. *Nucleic acids research* **43**, D234-239 (2015).

271. M. Tarailo-Graovac, N. Chen, Using RepeatMasker to identify repetitive elements in genomic sequences. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]* **Chapter 4**, Unit 4 10 (2009).

272. A. Rhoads, K. F. Au, PacBio Sequencing and Its Applications. *Genomics, proteomics & bioinformatics* **13**, 278-289 (2015).

273. S. B. Rothbart, B. D. Strahl, Interpreting the language of histone and DNA modifications. *Biochimica et biophysica acta* **1839**, 627-643 (2014).

274. E. J. Kim, X. Ma, H. Cerutti, Gene silencing in microalgae: mechanisms and biological roles. *Bioresource technology* **184**, 23-32 (2015).

275. I. V. Grigoriev *et al.*, The genome portal of the Department of Energy Joint Genome Institute. *Nucleic acids research* **40**, D26-32 (2012).

276. D. M. Goodstein *et al.*, Phytozome: a comparative platform for green plant genomics. *Nucleic acids research* **40**, D1178-1186 (2012).

277. E. Derelle *et al.*, Genome analysis of the smallest free-living eukaryote Ostreococcus tauri unveils many unique features. *Proc Natl Acad Sci U S A* **103**, 11647-11652 (2006).

278. P. S. Schnable *et al.*, The B73 maize genome: complexity, diversity, and dynamics. *Science* **326**, 1112-1115 (2009).

279. A. Gurevich, V. Saveliev, N. Vyahhi, G. Tesler, QUAST: quality assessment tool for genome assemblies. *Bioinformatics* **29**, 1072-1075 (2013).

280. K. F. Tipton, S. Boyce, History of the enzyme nomenclature system. *Bioinformatics* **16**, 34-40 (2000).

281. R. H. S. Thompson, Classification and nomenclature of enzymes and coenzymes. *Nature* **193**, 1227-1231 (1962).

282. A. Cornish-Bowden, Current IUBMB recommendations on enzyme nomenclature and kinetics. *Perspectives in Science* **1**, 74-87 (2014).

283. M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, M. Tanabe, KEGG as a reference resource for gene and protein annotation. *Nucleic acids research* **44**, D457-462 (2016).

284. T. G. O. Consortium, Gene Ontology: tool for the unification of biology. *Nature genetics* **25**, 25-29 (2000).

285. Z. Cai, X. Mao, S. Li, L. Wei, Genome comparison using Gene Ontology (GO) with statistical testing. *BMC bioinformatics* **7**, 374 (2006).

286. R. D. Finn *et al.*, The Pfam protein families database: towards a more sustainable future. *Nucleic acids research* **44**, D279-285 (2016).

287. K. Sheppard *et al.*, From one amino acid to another: tRNA-dependent amino acid biosynthesis. *Nucleic acids research* **36**, 1813-1825 (2008).

288. S. Takyar, R. P. Hickerson, H. F. Noller, mRNA helicase activity of the ribosome. *Cell* **120**, 49-58 (2005).

289. S. Sato, Y. Nakamura, T. Kaneko, E. Asamizu, S. Tabata, Complete Structure of the Chloroplast Genome of *Arabidopsis thaliana*. *DNA Res* **6**, 283-290 (1999).

290. M. Unseld, J. R. Marienfeld, P. Brandt, A. Brennicke, The mitochondrial genome of *Arabidopsis thaliana* contains 57 genes in 366,924 nucleotides. *Nature genetics* **15**, 57-61 (1997).

291. J. M. Lovgren *et al.*, The PRC-barrel domain of the ribosome maturation protein RimM mediates binding to ribosomal protein S19 in the 30S ribosomal subunits. *Rna* **10**, 1798-1812 (2004).

292. A. Elo, Nuclear Genes That Encode Mitochondrial Proteins for DNA and RNA Metabolism Are Clustered in the Arabidopsis Genome. *The Plant Cell Online* **15**, 1619-1631 (2003).

293. H. Gao *et al.*, Study of the Structural Dynamics of the E. coli 70S Ribosome Using Real-Space Refinement. *Cell* **113**, 789-801 (2003).

294. R. Hauser *et al.*, RsfA (YbeB) proteins are conserved ribosomal silencing factors. *PLoS genetics* **8**, e1002815 (2012).

295. A. P. Korepanov, A. V. Korobeinikova, S. A. Shestakov, M. B. Garber, G. M. Gongadze, Protein L5 is crucial for in vivo assembly of the bacterial 50S ribosomal subunit central protuberance. *Nucleic acids research* **40**, 9153-9159 (2012).

296. Y. Hirano, S. Murata, K. Tanaka, Large- and Small-Scale Purification of Mammalian 26S Proteasomes. *Methods in enzymology* **399**, 227-240 (2005).

297. M. Ueda *et al.*, The HALTED ROOT gene encoding the 26S proteasome subunit RPT2a is essential for the maintenance of Arabidopsis meristems. *Development* **131**, 2101-2111 (2004).

298. N. Qureshi *et al.*, The Proteasome as a Lipopolysaccharide-Binding Protein in Macrophages: Differential Effects of Proteasome Inhibition on Lipopolysaccharide-Induced Signaling Events. *The Journal of Immunology* **171**, 1515-1525 (2003).

299. K. Igarashi, A. Ishihama, Bipartite Functional Map of the E. coli RNA Polymerase alpha Subunit: Inolvement of the C-Terminal Region in Transcription Acivation by cAMP-CRP. *Cell* **65**, 1015-1022 (1991).

300. A. L. Chateigner-Boutin *et al.*, CLB19, a pentatricopeptide repeat protein required for editing of rpoA and clpP chloroplast transcripts. *The Plant journal : for cell and molecular biology* **56**, 590-602 (2008).

301. S. Steiner, Y. Schroter, J. Pfalz, T. Pfannschmidt, Identification of essential subunits in the plastid-encoded RNA polymerase complex reveals building blocks for proper plastid development. *Plant Physiol* **157**, 1043-1055 (2011).

302. Y. H. Chiu, J. B. Macmillan, Z. J. Chen, RNA polymerase III detects cytosolic DNA and induces type I interferons through the RIG-I pathway. *Cell* **138**, 576-591 (2009).

303. M. Werner, P. Thuriaux, J. Soutourina, Structure-function analysis of RNA polymerases I and III. *Current opinion in structural biology* **19**, 740-745 (2009).

304. E. Landrieux *et al.*, A subcomplex of RNA polymerase III subunits involved in transcription termination and reinitiation. *EMBO Journal* **25**, 118-128 (2006).

305. M. Renaud *et al.*, Gene duplication and neofunctionalization: POLR3G and POLR3GL. *Genome research* **24**, 37-51 (2014).

306. S. L. Sanders, K. A. Garbett, P. A. Weil, Molecular Characterization of Saccharomyces cerevisiae TFIID. *Molecular and cellular biology* **22**, 6000-6013 (2002).

307. C. S. Lutz, C. Cooke, J. P. O'Connor, R. Kobayashi, J. C. Alwine, The snRNP-free U1A (SF-A) complex(es): Identification of the largest subunit as PSF, the polypyrimidine-tract binding protein-associated splicing factor. *Rna* **4**, 1493-1499 (1998).

308. S. Sahara *et al.*, Acinus is a caspase-3-activated protein required for apoptotic chromatin condensation. *Nature* **401**, 168-173 (1999).

309. T. O. Tange, T. Shibuya, M. S. Jurica, M. J. Moore, Biochemical analysis of the EJC reveals two new factors and a stable tetrameric protein core. *Rna* **11**, 1869-1883 (2005).

310. M. P. Mayer, B. Bukau, Hsp70 chaperones: cellular functions and molecular mechanism. *Cellular and molecular life sciences : CMLS* **62**, 670-684 (2005).

311. A. S. Ma *et al.*, Heterogeneous nuclear ribonucleoprotein A3, a novel RNA trafficking response element-binding protein. *The Journal of biological chemistry* **277**, 18010-18020 (2002).

312. Q. Wang, B. C. Rymond, Rds3p Is Required for Stable U2 snRNP Recruitment to the Splicing Apparatus. *Molecular and cellular biology* **23**, 7339-7349 (2003).

313. J. R. Boyne, K. J. Colgan, A. Whitehouse, Recruitment of the complete hTREX complex is required for Kaposi's sarcoma-associated herpesvirus intronless mRNA nuclear export and virus replication. *PLoS Pathog* **4**, e1000194 (2008).

314. M. Ohno, Y. Shimura, A human RNA helicase-like protein, HRH1, facilitates nuclear export of spliced mRNA by releasing the RNA from the spliceosome. *Genes and Development* **10**, 997-1007 (1996).

315. R. Chen, M. S. Wold, Replication protein A: single-stranded DNA's first responder: dynamic DNA-interactions allow replication protein A to direct single-strand DNA intermediates into different pathways for synthesis or repair. *BioEssays : news and reviews in molecular, cellular and developmental biology* **36**, 1156-1161 (2014).

316. J. Kim *et al.*, Contrasting effects of Elg1-RFC and Ctf18-RFC inactivation in the absence of fully functional RFC in fission yeast. *Nucleic acids research* **33**, 4078-4089 (2005).

317. H. Li *et al.*, Functional roles of p12, the fourth subunit of human DNA polymerase delta. *The Journal of biological chemistry* **281**, 14748-14755 (2006).

318. J. H. Lee, T. T. Paull, Direct Activation of the ATM Protein Kinase by the Mre11/Rad50/Nbs1 Complex. *Science* **304**, 93-96 (2004).

319. A. V. Nimonkar *et al.*, BLM-DNA2-RPA-MRN and EXO1-BLM-RPA-MRN constitute two DNA end resection machineries for human DNA break repair. *Genes & development* **25**, 350-362 (2011).

320. L. C. Chuang *et al.*, Phosphorylation of Mcm2 by Cdc7 promotes pre-replication complex assembly during cell-cycle re-entry. *Mol Cell* **35**, 206-216 (2009).

321. M. D. Petroski, R. J. Deshaies, Function and regulation of cullin-RING ubiquitin ligases. *Nat Rev Mol Cell Biol* **6**, 9-20 (2005).

322. R. L. Ferguson, G. Pascreau, J. L. Maller, The cyclin A centrosomal localization sequence recruits MCM5 and Orc1 to regulate centrosome reduplication. *Journal of cell science* **123**, 2743-2749 (2010).

323. N. Arsic *et al.*, A novel function for Cyclin A2: control of cell invasion via RhoA signaling. *J Cell Biol* **196**, 147-162 (2012).

324. T. Miyazaki, S. Arai, Two Distinct Controls of Mitotic Cdk1/Cyclin B1 Activity Requisite for Cell Growth Prior to Cell Division. *Cell Cycle* **6**, 1418-1424 (2014).

325. B. Hopwood, S. Dalton, Cdc45p assemblies into a complex with Cdc46p/Mcm5p, is required for minichromosome maintenance, and is essential for chromosomal DNA replication. *PNAS* **93**, 12309-12314 (1996).

326. J. Li *et al.*, Phosphorylation of MCM3 protein by cyclin E/cyclin-dependent kinase 2 (Cdk2) regulates its function in cell cycle. *The Journal of biological chemistry* **286**, 39776-39785 (2011).

327. T. Asano, M. Makise, M. Takehara, T. Mizushima, Interaction between ORC and Cdt1p of Saccharomyces cerevisiae. *FEMS Yeast Res* **7**, 1256-1262 (2007).

328. L. S. van Bezouwen *et al.*, Subunit and chlorophyll organization of the plant photosystem II supercomplex. *Nat Plants* **3**, 17080 (2017).

329. D. A. Berthold, C. L. Schmidt, R. Malkin, The Deletion of *petG* in *Chlamydomonas reinhardtii* Disrupts the Cytochrome *bf* Complex. *The Journal of biological chemistry* **270**, 29293-29298 (1995).

330. J. M. Mach, A. R. Castillo, R. Hoogstraten, J. T. Greenberg, The Arabidopsis-accelerated cell death gene ACD2 encodes red chlorophyll catabolite reductase and suppresses the spread of disease symptoms. *Proc Natl Acad Sci U S A* **98**, 771-776 (2001).

331. N. Frankenberg, K. Mukougawa, T. Kohchi, J. C. Lagarias, Functional Genomic Analysis of the HY2 Family of Ferredoxin-Dependent Bilin Reductases from Oxygenic Photosynthetic Organisms. *The Plant cell* **13**, 965-978 (2001).

332. T. Tsuchiya *et al.*, Cloning of chlorophyllase, the key enzyme in chlorophyll degradation: finding of a lipase motif and the induction by methyl jasmonate. *Proc Natl Acad Sci U S A* **96**, 15362-15367 (1999).

333. M. Wikstrom, G. Hummer, Stoichiometry of proton translocation by respiratory complex I and its mechanistic implications. *Proc Natl Acad Sci U S A* **109**, 4431-4436 (2012).

334. A. Guimier *et al.*, Biallelic PPA2 Mutations Cause Sudden Unexpected Cardiac Arrest in Infancy. *Am J Hum Genet* **99**, 666-673 (2016).

335. J. J. Salas, J. B. Ohlrogge, Characterization of substrate specificity of plant FatA and FatB acyl-ACP thioesterases. *Arch Biochem Biophys* **403**, 25-34 (2002).

336. C. de Azevedo Souza *et al.*, A novel fatty Acyl-CoA Synthetase is required for pollen development and sporopollenin biosynthesis in Arabidopsis. *The Plant cell* **21**, 507-525 (2009).

337. J. Schnurr, J. Shockey, J. Browse, The acyl-CoA synthetase encoded by LACS2 is essential for normal cuticle development in Arabidopsis. *The Plant cell* **16**, 629-642 (2004).

338. I. P. Pulsifer, S. Kluge, O. Rowland, Arabidopsis long-chain acyl-CoA synthetase 1 (LACS1), LACS2, and LACS3 facilitate fatty acid uptake in yeast. *Plant physiology and biochemistry : PPB / Societe francaise de physiologie vegetale* **51**, 31-39 (2012).

339. H. Inoue, H. Sagami, T. Koyama, K. Ogura, Properties of farnesol phosphokinase of *Botryococcus braunii*. *Phytochemistry* **40**, 377-381 (1995).

340. S. K. Oh, K. H. Han, S. B. Ryu, H. Kang, Molecular cloning, expression, and functional analysis of a cis-prenyltransferase from Arabidopsis thaliana. Implications in rubber biosynthesis. *The Journal of biological chemistry* **275**, 18482-18488 (2000).

341. L. Stange, E. L. Bennett, M. Calvin, Short-time 14CO2 incorporation experiments with synchronously growing *Chlorella* cells. *Biochimica et biophysica acta* **37**, 78-92 (1960).

342. S. Surzycki, Synchronously Grown Cultures of Chlarnydomonas reinhardi. *Methods in enzymology* **23**, 67-73 (1971).

343. K. Krupinska, K. Humbeck, Light-induced synchronous cultures, an excellent tool to study the cell cycle of unicellular green algae. *J Photochem Photobiol B* **26**, 217-231 (1994).

344. M. Mittag, Circadian Rhythms in Microalgae. *Int Rev Cyt* **206**, 213-247 (2001).

345. S. Cooper, Rejoinder: whole-culture synchronization cannot, and does not, synchronize cells. *Trends in biotechnology* **22**, 274-276 (2004).

346. E. M. Farre, The regulation of plant growth by the circadian clock. *Plant biology* **14**, 401-410 (2012).

347. M. Sorek, O. Levy, The effect of temperature compensation on the circadian rhythmicity of photosynthesis in Symbiodinium, coral-symbiotic alga. *Scientific reports* **2**, 536 (2012).

348. M. Sorek, O. Levy, Influence of the quantity and quality of light on photosynthetic periodicity in coral endosymbiotic algae. *PLoS One* **7**, e43264 (2012).

349. S. Y. Miyagishima *et al.*, Translation-independent circadian control of the cell cycle in a unicellular photosynthetic eukaryote. *Nature communications* **5**, 3807 (2014).

350. S. Diamond, D. Jun, B. E. Rubin, S. S. Golden, The circadian oscillator in Synechococcus elongatus controls metabolite partitioning during diurnal growth. *Proc Natl Acad Sci U S A* **112**, E1916-1925 (2015).

351. Z. B. Noordally, A. J. Millar, Clocks in algae. *Biochemistry* **54**, 171-183 (2015).

352. F. R. Cross, J. G. Umen, The Chlamydomonas cell cycle. *The Plant Journal* **82**, 370-392 (2015).

353. N. Panchy *et al.*, Prevalence, evolution, and cis-regulation of diel transcription in Chlamydomonas reinhardtii. *G3 (Bethesda)* **4**, 2461-2471 (2014).

354. J. M. Zones, I. K. Blaby, S. S. Merchant, J. G. Umen, High-Resolution Profiling of a Synchronized Diurnal Transcriptome from Chlamydomonas reinhardtii Reveals Continuous Cell and Metabolic Differentiation. *The Plant cell* **27**, 2743-2769 (2015).

355. E. Poliner *et al.*, Transcriptional coordination of physiological responses in Nannochloropsis oceanica CCMP1779 under light/dark cycles. *The Plant journal : for cell and molecular biology* **83**, 1097-1113 (2015).

356. P. de Los Reyes, F. J. Romero-Campero, M. T. Ruiz, J. M. Romero, F. Valverde, Evolution of Daily Gene Co-expression Patterns from Algae to Plants. *Frontiers in plant science* **8**, 1217 (2017).

357. N. L. Bray, H. Pimentel, P. Melsted, L. Pachter, Near-optimal probabilistic RNA-seq quantification. *Nat Biotechnol* **34**, 525-527 (2016).

358. B. J. Haas *et al.*, De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nature protocols* **8**, 1494-1512 (2013).

359. C. W. Law, Y. Chen, W. Shi, G. K. Smyth, voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol* **15**, R29 (2014).

360. M. D. Robinson, D. J. McCarthy, G. K. Smyth, edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139-140 (2010).

361. M. I. Love, W. Huber, S. Anders, Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* **15**, 550 (2014).

362. N. J. Schurch *et al.*, How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *Rna* **22**, 839-851 (2016).

363. C. Liseron-Monfils, D. Ware, Revealing gene regulation and associations through biological networks. *Current Plant Biology* **3-4**, 30-39 (2015).

364. M. Tatli, M. T. Naik, S. Okada, L. J. Dangott, T. P. Devarenne, Isolation and Characterization of Cyclic C33 Botryococcenes and a Trimethylsqualene Isomer from Botryococcus braunii Race B. *J Nat Prod*, (2017).

365. D. Nashta-ali, S. A. Motahari, B. Hossein Khalaj, Breaking Lander-Waterman's Coverage Bound. (2016).

366. J. L. Fierst, D. A. Murdock, Decontaminating eukaryotic genome assemblies with machine learning. *BMC bioinformatics* **18**, 533 (2017).

367. M. W. Libbrecht, W. S. Noble, Machine learning applications in genetics and genomics. *Nature reviews. Genetics* **16**, 321-332 (2015).

368. L. Hou, H. Park, S. Okada, T. Ohama, Release of single cells from the colonial oil-producing alga Botryococcus braunii by chemical treatments. *Protoplasma*, (2013).

369. S. E. Prochnik *et al.*, Genomic analysis of organismal complexity in the multicellular green alga Volvox carteri. *Science* **329**, 223-226 (2010).

370. R. Carlson, Estimating the biotech sector's contribution to the US economy. *Nat Biotechnol* **34**, 247-255 (2016).

371. N. J. Kelley *et al.*, Engineering Biology to Address Global Problems: Synthetic Biology Markets, Needs, and Applications. *Industrial Biotechnology* **10**, 140-149 (2014).

372. T. Vasconcelos Fernandes *et al.*, Closing Domestic Nutrient Cycles Using Microalgae. *Environ Sci Technol* **49**, 12450-12456 (2015).

373. V. H. Work *et al.*, Biocommodities from photosynthetic microorganisms. *Environmental Progress & Sustainable Energy*, n/a-n/a (2013).

374. M. G. Grabherr *et al.*, Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat Biotechnol* **29**, 644-652 (2011).

375. B. J. Haas, Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies. *Nucleic acids research* **31**, 5654-5666 (2003).

376. G. S. Slater, E. Birney, Automated generation of heuristics for biological sequence comparison. *BMC bioinformatics* **6**, 31 (2005).

377. P. J. Keeling *et al.*, The Marine Microbial Eukaryote Transcriptome Sequencing Project (MMETSP): illuminating the functional diversity of eukaryotic life in the oceans through transcriptome sequencing. *PLoS Biol* **12**, e1001889 (2014).

378. C. The UniProt, UniProt: the universal protein knowledgebase. *Nucleic acids research* **45**, D158-D169 (2017).

379. A. Bairoch, B. Boeckmann, The SWISS-PROT protein sequence data bank: curent status. *Nucleic acids research* **22**, 3578-3580 (1994).

380. A. Salamov, V. Solovyev, Ab initio Gene Finding in *Drosophila* Genomic DNA. *Genome research* **10**, 516-522 (2000).

381. R. F. Yeh, L. P. Lim, C. B. Burge, Computational Inference of Homologous Gene Structures in the Human Genome. *Genome research* **11**, 803-816 (2001).

382. K. J. Hoff, S. Lange, A. Lomsadze, M. Borodovsky, M. Stanke, BRAKER1: Unsupervised RNA-Seq-Based Genome Annotation with GeneMark-ET and AUGUSTUS. *Bioinformatics* **32**, 767-769 (2016).

383. R. D. Finn *et al.*, InterPro in 2017-beyond protein family and domain annotations. *Nucleic acids research* **45**, D190-D199 (2017).

384. P. Jones *et al.*, InterProScan 5: genome-scale protein function classification. *Bioinformatics* **30**, 1236-1240 (2014).

385. L. Chae, T. Kim, R. Nilo-Poyanco, S. Y. Rhee, Genomic Signatures of Specialized Metabolism in Plants. *Science* **344**, 510-513 (2014).

386. M. Grung, P. Metzger, S. Liaaen-Jensen, Primary and Secondary Carotenoids in Two Races of the Green Alga *Botryococcus braunii*. *Biochemical Systematics and Ecology* **17**, 263-269 (1989).

APPENDIX A

SUPPLEMENTARY MATERIAL FOR THE GENOME OF *BOTRYOCOCCUS BRAUNII*

A.1 Materials and Methods for Testing Assembly of the *B. braunii* Genome

The Illumina HiSeq 2500 library SXPX was utilized as the basis for testing different methods of assembly. This library is available from the JGI Genome Portal (https://genome.jgi.doe.gov/portal/) under JGI Project ID 1014520. It is also available from the NCBI Sequence Read Archive (https://www.ncbi.nlm.nih.gov/sra) under accession ID SRX2043336. The raw reads from these two databases are stored in a compressed FASTQ file with interleaved pairs. Thus the first operation after downloading the reads is to decompress and unweave the pairs into separate files for the forward (left) and reverse (right) reads. This was achieved by using a script included in the Trinity RNA-seq software suite, as follows:

```
$ zcat ../01_Interleaved_Pairs/Library.SXPX.7901.1.86132.GAGTGG.anqdp.fastq.gz |
/scratch/user/dbrowne/Software/trinityrnaseq-2.0.6-westmere/util/misc/fastq_unweave_pairs.pl -
Library.SXPX.L.fq Library.SXPX.R.fq &
```

A.1.1 Combining Multiple de Bruijn Graph Assemblies

After the reads of library SXPX were unweaved and stored into separate files for the left and right reads, each of these two files was split into 125 smaller files by using the following commands:

```
$ (zcat ../../Library.SXPX.L.fq.gz | split -l 8000000 - SXPX.L_2M) && gzip -S .fq.gz *;
$ (zcat ../../Library.SXPX.R.fq.gz | split -l 8000000 - SXPX.R_2M) && gzip -S .fq.gz *;
```

The assembler used for this stage of the experiment was ABYSS-P, the parallelized version of the ABYSS assembler. This program was selected because it could scale across the Ada supercomputer, taking advantage of the many available nodes through an MPI-based design,

enabling quick assembly of the data. The split SXPX file names were collected into a text file and then re-processed using the following Python commands:

```
file_list = open('SXPX_PE_Files.txt', 'rU').read().split('\n')
del file_list[-1]
output = open("SXPX_Formatted_for_ABySS.txt", "a")
for i in range(len(file_list)):
    print >>output, 'pe'+str(i+1),
for i in range(len(file_list)):
    file_list[i] = file_list[i].split('\t')
    print >>output, 'pe'+str(i+1)+'="SXPX_SPLIT_2M/PE_L/'+file_list[i][0]+'
SXPX_SPLIT_2M/PE_R/'+file_list[i][1]+'"',
```

This yielded part of the required ABYSS-P command, which needed all 250 of the file names. The output from the above commands were plugged into a template ABYSS command:

```
$ abyss-pe -n v=-v k=KMER name=NAME s=500 np=250 j=5 lib="pe1 pe2 pe3 pe4 pe5 pe6 pe7 pe8 pe9
pe10 pe11 pe12 pe13 pe14 pe15 pe16 pe17 pe18 pe19 pe20 pe21 pe22 pe23 pe24 pe25 pe26 pe27 pe28
pe29 pe30 pe31 pe32 pe33 pe34 pe35 pe36 pe37 pe38 pe39 pe40 pe41 pe42 pe43 pe44 pe45 pe46 pe47
pe48 pe49 pe50 pe51 pe52 pe53 pe54 pe55 pe56 pe57 pe58 pe59 pe60 pe61 pe62 pe63 pe64 pe65 pe66
pe67 pe68 pe69 pe70 pe71 pe72 pe73 pe74 pe75 pe76 pe77 pe78 pe79 pe80 pe81 pe82 pe83 pe84 pe85
pe86 pe87 pe88 pe89 pe90 pe91 pe92 pe93 pe94 pe95 pe96 pe97 pe98 pe99 pe100 pe101 pe102 pe103
pe104 pe105 pe106 pe107 pe108 pe109 pe110 pe111 pe112 pe113 pe114 pe115 pe116 pe117 pe118
pe119 pe120 pe121 pe122 pe123 pe124 pe125" pe1="SXPX_SPLIT_2M/PE_L/SXPX.L_2Maa.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Maa.fq.gz" pe2="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mab.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mab.fq.gz" pe3="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mac.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mac.fq.gz" pe4="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mad.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mad.fq.gz" pe5="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mae.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mae.fq.gz" pe6="SXPX_SPLIT_2M/PE_L/SXPX.L_2Maf.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Maf.fq.gz" pe7="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mag.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mag.fq.gz" pe8="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mah.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mah.fq.gz" pe9="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mai.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mai.fq.gz" pe10="SXPX_SPLIT_2M/PE_L/SXPX.L_2Maj.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Maj.fq.gz" pe11="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mak.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mak.fq.gz" pe12="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mal.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mal.fq.gz" pe13="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mam.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mam.fq.gz" pe14="SXPX_SPLIT_2M/PE_L/SXPX.L_2Man.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Man.fq.gz" pe15="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mao.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mao.fq.gz" pe16="SXPX_SPLIT_2M/PE_L/SXPX.L_2Map.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Map.fq.gz" pe17="SXPX_SPLIT_2M/PE_L/SXPX.L_2Maq.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Maq.fq.gz" pe18="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mar.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mar.fq.gz" pe19="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mas.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mas.fq.gz" pe20="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mat.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mat.fq.gz" pe21="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mau.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mau.fq.gz" pe22="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mav.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mav.fq.gz" pe23="SXPX_SPLIT_2M/PE_L/SXPX.L_2Maw.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Maw.fq.gz" pe24="SXPX_SPLIT_2M/PE_L/SXPX.L_2Max.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Max.fq.gz" pe25="SXPX_SPLIT_2M/PE_L/SXPX.L_2May.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2May.fq.gz" pe26="SXPX_SPLIT_2M/PE_L/SXPX.L_2Maz.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Maz.fq.gz" pe27="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mba.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mba.fq.gz" pe28="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbb.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbb.fq.gz" pe29="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbc.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbc.fq.gz" pe30="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbd.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbd.fq.gz" pe31="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbe.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbe.fq.gz" pe32="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbf.fq.gz
```

```
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbf.fq.gz" pe33="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbg.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbg.fq.gz" pe34="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbh.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbh.fq.gz" pe35="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbi.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbi.fq.gz" pe36="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbj.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbj.fq.gz" pe37="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbk.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbk.fq.gz" pe38="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbl.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbl.fq.gz" pe39="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbm.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbm.fq.gz" pe40="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbn.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbn.fq.gz" pe41="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbo.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbo.fq.gz" pe42="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbp.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbp.fq.gz" pe43="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbq.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbq.fq.gz" pe44="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbr.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbr.fq.gz" pe45="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbs.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbs.fq.gz" pe46="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbt.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbt.fq.gz" pe47="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbu.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbu.fq.gz" pe48="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbv.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbv.fq.gz" pe49="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbw.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbw.fq.gz" pe50="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbx.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbx.fq.gz" pe51="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mby.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mby.fq.gz" pe52="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mbz.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mbz.fq.gz" pe53="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mca.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mca.fq.gz" pe54="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcb.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcb.fq.gz" pe55="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcc.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcc.fq.gz" pe56="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcd.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcd.fq.gz" pe57="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mce.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mce.fq.gz" pe58="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcf.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcf.fq.gz" pe59="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcg.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcg.fq.gz" pe60="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mch.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mch.fq.gz" pe61="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mci.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mci.fq.gz" pe62="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcj.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcj.fq.gz" pe63="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mck.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mck.fq.gz" pe64="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcl.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcl.fq.gz" pe65="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcm.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcm.fq.gz" pe66="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcn.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcn.fq.gz" pe67="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mco.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mco.fq.gz" pe68="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcp.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcp.fq.gz" pe69="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcq.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcq.fq.gz" pe70="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcr.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcr.fq.gz" pe71="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcs.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcs.fq.gz" pe72="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mct.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mct.fq.gz" pe73="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcu.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcu.fq.gz" pe74="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcv.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcv.fq.gz" pe75="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcw.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcw.fq.gz" pe76="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcx.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcx.fq.gz" pe77="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcy.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcy.fq.gz" pe78="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mcz.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mcz.fq.gz" pe79="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mda.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mda.fq.gz" pe80="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdb.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdb.fq.gz" pe81="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdc.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdc.fq.gz" pe82="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdd.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdd.fq.gz" pe83="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mde.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mde.fq.gz" pe84="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdf.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdf.fq.gz" pe85="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdg.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdg.fq.gz" pe86="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdh.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdh.fq.gz" pe87="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdi.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdi.fq.gz" pe88="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdj.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdj.fq.gz" pe89="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdk.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdk.fq.gz" pe90="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdl.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdl.fq.gz" pe91="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdm.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdm.fq.gz" pe92="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdn.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdn.fq.gz" pe93="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdo.fq.gz
```

```
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdo.fq.gz" pe94="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdp.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdp.fq.gz" pe95="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdq.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdq.fq.gz" pe96="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdr.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdr.fq.gz" pe97="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mds.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mds.fq.gz" pe98="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdt.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdt.fq.gz" pe99="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdu.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdu.fq.gz" pe100="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdv.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdv.fq.gz" pe101="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdw.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdw.fq.gz" pe102="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdx.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdx.fq.gz" pe103="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdy.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdy.fq.gz" pe104="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mdz.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mdz.fq.gz" pe105="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mea.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mea.fq.gz" pe106="SXPX_SPLIT_2M/PE_L/SXPX.L_2Meb.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Meb.fq.gz" pe107="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mec.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mec.fq.gz" pe108="SXPX_SPLIT_2M/PE_L/SXPX.L_2Med.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Med.fq.gz" pe109="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mee.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mee.fq.gz" pe110="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mef.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mef.fq.gz" pe111="SXPX_SPLIT_2M/PE_L/SXPX.L_2Meg.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Meg.fq.gz" pe112="SXPX_SPLIT_2M/PE_L/SXPX.L_2Meh.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Meh.fq.gz" pe113="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mei.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mei.fq.gz" pe114="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mej.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mej.fq.gz" pe115="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mek.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mek.fq.gz" pe116="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mel.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mel.fq.gz" pe117="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mem.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mem.fq.gz" pe118="SXPX_SPLIT_2M/PE_L/SXPX.L_2Men.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Men.fq.gz" pe119="SXPX_SPLIT_2M/PE_L/SXPX.L_2Meo.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Meo.fq.gz" pe120="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mep.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mep.fq.gz" pe121="SXPX_SPLIT_2M/PE_L/SXPX.L_2Meq.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Meq.fq.gz" pe122="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mer.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mer.fq.gz" pe123="SXPX_SPLIT_2M/PE_L/SXPX.L_2Mes.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Mes.fq.gz" pe124="SXPX_SPLIT_2M/PE_L/SXPX.L_2Met.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Met.fq.gz" pe125="SXPX_SPLIT_2M/PE_L/SXPX.L_2Meu.fq.gz
SXPX_SPLIT_2M/PE_R/SXPX.R_2Meu.fq.gz"
```

The above command produced a “dry run” output that yielded the commands from the ABySS pipeline that needed to be run in order to assemble the data. A template file was created to execute assembly with different k-mer values. The template was distributed with a small shell scripts:

```
for i in {50..250}; do
    cp Stage_1_Template.job ./Job_Scripts/k$i/k$i\_1.job
    sed -i "s/KMER/$i/g" ./Job_Scripts/k$i/k$i\_1.job
    sed -i "s/NAME/SXPX_k$i/g" ./Job_Scripts/k$i/k$i\_1.job
done
```

This script yielded a set of job files for submission to the LSF batch processing system. These job files were submitted with a shell script as follows:

```
for i in {50..250}; do
    bsub < Job_Scripts/k$i/k$i\_1.job
```

The result from these jobs was a set of ABYSS-P assemblies, generated from k-mers between 50 and 200, summarized in Figure 3. The next step was to consolidate these assemblies into a set of de-replicated contigs. This was accomplished using the CD-HIT program. First, all of the ABYSS-P assemblies were collected into a single FASTA file as follows:

```
$ find ../2015.12.01_ABySS_k50-250_SXPX/Working_Folders/ -name *-1.fa | xargs cat >
Total_Assemblies-1.fa
```

The total set of sequences contained contigs as small as 50 bp, and thus in order to simplify the dataset, two sub-sets of the total contigs were created, with minimum length thresholds of 300 bp and 1,000 bp. These two sub-sets were created as follows:

```
$ awk '!/^>/ { next } { getline seq } length(seq) >= 1000 { print $0 "\n" seq }'
Total_Assemblies-1.fa > Total_Assemblies_SEQ1K.fa
$ awk '!/^>/ { next } { getline seq } length(seq) >= 300 { print $0 "\n" seq }'
Total_Assemblies-1.fa
```

Redundancy amongst the contigs in these two sets was eliminated with CD-HIT by issuing the following commands:

```
$ cd-hit-est -i ../Total_Assemblies_SEQ1K.fa -o SXPX_SEQ1K_95 -c 0.95 -T 20 -M 53000
$ cd-hit-est -i ../Total_Assemblies_SEQ1K.fa -o SXPX_SEQ1K_99 -c 0.99 -T 20 -M 53000
$ cd-hit-est -i ../Total_Assemblies_SEQ1K.fa -o SXPX_SEQ1K_100 -c 1 -T 20 -M 53000
$ cd-hit-est -i ../Total_Assemblies_SEQ300.fa -o SXPX_SEQ300_95 -c 0.95 -T 20 -M 53000
$ cd-hit-est -i ../Total_Assemblies_SEQ300.fa -o SXPX_SEQ300_99 -c 0.99 -T 20 -M 53000
$ cd-hit-est -i ../Total_Assemblies_SEQ300.fa -o SXPX_SEQ300_100 -c 1 -T 20 -M 53000
```

The set of consolidated contigs called SEQ1K_100 was further processed with tools from ABySS to yield an OLC assembly. The following commands were issued:

```
$ abyss-fac -s 0 SXPX_SEQ1K_100-1.fa
$ abyss-overlap --adj -m 100 -j 20 --no-SS -v SXPX_SEQ1K_100-1.fa > SXPX_SEQ1K_100-1.dot
$ abyss-filtergraph -v -k 1000 -m 100 --dot --no-SS --no-shim --assemble -g SXPX_SEQ1K_100-2.dot
SXPX_SEQ1K_100-1.dot SXPX_SEQ1K_100-1.fa > SXPX_SEQ1K_100-2.path
$ MergeContigs -v -k 1000 --adj -g SXPX_SEQ1K_100-3.dot -o SXPX_SEQ1K_100-3.fa SXPX_SEQ1K_100-
1.fa SXPX_SEQ1K_100-2.dot SXPX_SEQ1K_100-2.path
```

The results of the CD-HIT consolidation are presented in Table 3 and the results of the OLC re-assembly are presented in Table 4.

A.1.2 Assembling Illumina Data with DISCOVAR de novo

The DISCOVAR de novo program (version 52488) was used to assemble the library SXPX separated, unsplit pairs. The following command was issued:

```
$ DiscovarDeNovo READS=Library.SXPX.L.fq.gz,Library.SXPX.R.fq.gz OUT_DIR=SXPX_DDN
NUM_THREADS=40 MAX_MEM_GB=800
```

The resulting assembly was renamed "B_DDN3_BASE-1.fa" and processed with tools from ABySS as follows:

```
$ abyss-overlap -v -m199 -k200 -j5 --no-SS B_DDN3_BASE-1.fa > B_DDN3_BASE-1.dot
$ abyss-filtergraph --no-SS --no-shim --gfa -k200 -v --assemble -t2000 B_DDN3_BASE-1.dot -g
B_DDN3_BASE-2.gfa > B_DDN3_BASE-2.path
```

The resulting GFA file "B_DDN3_BASE-2.gfa" was visualized with the Bandage program (version 0.8.1), showing the structure of the assembly graphs that were produced by DISCOVAR de novo, and presented in Figure 6.

A.1.3 Scaffolding and Gap Filling the DISCOVAR Assembly

The three libraries SXPX, NGNB, and HOOW were aligned against the DISCOVAR de novo assembly with BWA using the following commands:

```
$ bwa mem -t 20 -k 20 SXPX_DDN_v5-unitigs.fa 02_Separated_Pairs/Library.HOOW.L.fq.gz
02_Separated_Pairs/Library.HOOW.R.fq.gz | samtools view -@ 20 -Sb - | samtools sort -@ 20 -O
bam -T BWA_HOOW > BWA_HOOW.bam && samtools index BWA_HOOW.bam
$ bwa mem -t 20 -k 20 SXPX_DDN_v5-unitigs.fa 02_Separated_Pairs/Library.NGNB.L.fq.gz
02_Separated_Pairs/Library.NGNB.R.fq.gz | samtools view -@ 20 -Sb - | samtools sort -@ 20 -O
bam -T BWA_NGNB > BWA_NGNB.bam && samtools index BWA_NGNB.bam
$ bwa mem -t 20 -k 30 SXPX_DDN_v5-unitigs.fa 02_Separated_Pairs/Library.SXPX.L.fq.gz
02_Separated_Pairs/Library.SXPX.R.fq.gz | samtools view -@ 20 -Sb - | samtools sort -@ 20 -O
bam -T BWA_SXPX > BWA_SXPX.bam && samtools index BWA_SXPX.bam
```

The resulting BAM alignment files were given, along with the contigs, to BESST for scaffolding, using the following command:

```
$ runBESST -c SXPX_DDN_v5-unitigs.fa -f BWA_SXPX.bam BWA_NGNB.bam BWA_HOOW.bam -o
./BESST_OUTPUT_v1 --orientation fr rf rf --iter 1500000 -plots --separate_repeats --min_mapq 40
--dfs_traversal --no_score
```

Gaps in the scaffolds were filled with PBJelly, which consists of a complex pipeline of commands, coordinated by an XML protocol, as follows:

```
<jellyProtocol>
    <reference>/scratch/user/dbrowne/2016.05_MAY/2016.05.18_Race.B_PBJelly_Round_2/
DDN_BESST.fasta</reference>
    <outputDir>/scratch/user/dbrowne/2016.05_MAY/2016.05.18_Race.B_PBJelly_Round_2/
Jelly_Output_v2/</outputDir>
    <cluster>
        <command notes="For single node, multi-core machines" >${CMD} ${JOBNAME} 2> ${STDERR}
1> ${STDOUT} &amp;</command>
        <nJobs>1</nJobs>
    </cluster>
    <blasr>-minMatch 12 -maxMatch 250 -minPctSimilarity 70 -bestn 5 -nCandidates 10 -maxScore -
500 -nproc 20 -noSplitSubreads</blasr>
    <input baseDir="/scratch/user/dbrowne/2016.05_MAY/2016.05.18_Race.B_PBJelly_Round_2/">
        <job>PACBIO_Reads_3kb_Min.fasta</job>
    </input>
</jellyProtocol>
```

The following commands were issued to execute the indicated stages of PBJelly:

```
$ Jelly.py mapping Jelly_Protocol_v2.xml
$ Jelly.py support Jelly_Protocol_v2.xml
$ Jelly.py extraction Jelly_Protocol_v2.xml
$ Jelly.py assembly Jelly_Protocol_v2.xml -x "--nproc=20"
$ Jelly.py output Jelly_Protocol_v2.xml
```

After gap filling, the libraries SXPX, NGNB, and HOOW were aligned against the gap-filled assembly with BWA, using the same commands as before.

A.1.4 Assembling the PacBio Data with FALCON and ABruijn

The FALCON assemblies were performed at the HudsonAlpha Institute for Biotechnology. No detailed information on methods or protocols was included with the delivery of the assemblies.

The ABruijn assemblies were performed with a FASTA version of the PacBio data, with all reads shorter than either 6 kb or 10 kb in length removed from the dataset. The following commands were issued to initiate ABruijn:

```
$ abruijn.py -t 40 -i 3 -o 3000 PACBIO_Reads_6kb_Min.fasta BbB_ABruijn_v5 105
$ abruijn.py -t 40 -i 3 -o 3000 PACBIO_Reads_10kb_Min.fasta BbB_ABruijn_v4 43
```

In order to determine the impact of k-mer size on the resulting assembly, a range of values from 10-20 was tested. However, the only values that did not break the program and yielded assemblies were 12-16. The following commands were issued to obtain the assemblies:

```
$ abruijn.py -k 12 -t 20 -i 3 -o 3000 ../../PACBIO_Reads_6kb_Min.fasta ABKO_k12 105
$ abruijn.py -k 13 -t 20 -i 3 -o 3000 ../../PACBIO_Reads_6kb_Min.fasta ABKO_k12 105
$ abruijn.py -k 14 -t 20 -i 3 -o 3000 ../../PACBIO_Reads_6kb_Min.fasta ABKO_k12 105
$ abruijn.py -k 15 -t 20 -i 3 -o 3000 ../../PACBIO_Reads_6kb_Min.fasta ABKO_k12 105
$ abruijn.py -k 16 -t 20 -i 3 -o 3000 ../../PACBIO_Reads_6kb_Min.fasta ABKO_k12 105
```

A.1.5 Comparing ABYSS, DISCOVAR, FALCON, and ABruijn Assemblies

The four assemblies were compared at k-mers of 15, 20, 25, and 1000, using the K-mer Analysis Toolkit (KAT) software package version 2.1.1 (https://github.com/TGAC/KAT). The following commands were issued to obtain pairwise comparisons of the four assemblies at each of the different k-mers:

```
$ kat comp -v -m15 -n -t40 -o ABY_DDN_k15 ../B_ABYSS_K1000-1.fa ../B_DDN4_BASE-7.fa
$ kat comp -v -m15 -n -t40 -o ABY_ABR_k15 ../B_ABYSS_K1000-1.fa ../polished_3.fasta
$ kat comp -v -m15 -n -t40 -o ABY_FAL_k15 ../B_ABYSS_K1000-1.fa
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m15 -n -t40 -o DDN_ABR_k15 ../B_DDN4_BASE-7.fa ../polished_3.fasta
$ kat comp -v -m15 -n -t40 -o DDN_FAL_k15 ../B_DDN4_BASE-7.fa
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m15 -n -t40 -o ABR_FAL_k15 ../polished_3.fasta
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m20 -n -t40 -o ABY_DDN_k20 ../B_ABYSS_K1000-1.fa ../B_DDN4_BASE-7.fa
$ kat comp -v -m20 -n -t40 -o ABY_ABR_k20 ../B_ABYSS_K1000-1.fa ../polished_3.fasta
$ kat comp -v -m20 -n -t40 -o ABY_FAL_k20 ../B_ABYSS_K1000-1.fa
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m20 -n -t40 -o DDN_ABR_k20 ../B_DDN4_BASE-7.fa ../polished_3.fasta
$ kat comp -v -m20 -n -t40 -o DDN_FAL_k20 ../B_DDN4_BASE-7.fa
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m20 -n -t40 -o ABR_FAL_k20 ../polished_3.fasta
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m25 -n -t40 -o ABY_DDN_k25 ../B_ABYSS_K1000-1.fa ../B_DDN4_BASE-7.fa
$ kat comp -v -m25 -n -t40 -o ABY_ABR_k25 ../B_ABYSS_K1000-1.fa ../polished_3.fasta
```

```
$ kat comp -v -m25 -n -t40 -o ABY_FAL_k25 ../B_ABYSS_K1000-1.fa
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m25 -n -t40 -o DDN_ABR_k25 ../B_DDN4_BASE-7.fa ../polished_3.fasta
$ kat comp -v -m25 -n -t40 -o DDN_FAL_k25 ../B_DDN4_BASE-7.fa
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m25 -n -t40 -o ABR_FAL_k25 ../polished_3.fasta
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m1000 -n -t40 -o ABY_DDN_k1000 ../B_ABYSS_K1000-1.fa ../B_DDN4_BASE-7.fa
$ kat comp -v -m1000 -n -t40 -o ABY_ABR_k1000 ../B_ABYSS_K1000-1.fa ../polished_3.fasta
$ kat comp -v -m1000 -n -t40 -o ABY_FAL_k1000 ../B_ABYSS_K1000-1.fa
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m1000 -n -t40 -o DDN_ABR_k1000 ../B_DDN4_BASE-7.fa ../polished_3.fasta
$ kat comp -v -m1000 -n -t40 -o DDN_FAL_k1000 ../B_DDN4_BASE-7.fa
../polished_BOT6.NM2kb.fixed.full.fasta
$ kat comp -v -m1000 -n -t40 -o ABR_FAL_k1000 ../polished_3.fasta
../polished_BOT6.NM2kb.fixed.full.fasta
```

The data from these commands were manually processed into an Excel spreadsheet consisting of pairwise matrices. The matrix data for each comparison were then visualized with Python using the pandas, matplotlib, and Seaborn libraries, as in the following example:

```
import pandas as pd
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt

raw = """1    0.833204932   0.624970272   0.659583224
0.833204932   1      0.593435286   0.632866396
0.624970272   0.593435286   1      0.741865197
0.659583224   0.632866396   0.741865197   1"""
processed = raw.split('\n')
processed = [x.split('\t') for x in processed]
processed = [[float(i) for i in x] for x in processed]
Index = Cols = ['ABY', 'DDN', 'ABR', 'FAL']
df = pd.DataFrame(processed, index=Index, columns=Cols)
sns.heatmap(df, vmin=0, vmax=1, cmap=plt.cm.Blues)
plt.savefig('NGS_Comparison_k20.png')
plt.clf()
```

To count 1,000-mers in the four *B. braunii* genome assemblies (ABYSS, FALCON, DISCOVAR, and ABruijn), the KAT software was employed. The following commands were issued:

```
$ kat hist -t 40 -h 100000 -m 1000 -o kat_hist_k1000 ../BbB_mainGenome.fasta
$ kat hist -t 40 -H 165000000 -m 1000 -o kat_hist_ABv5_k10K B_ABv5-3.fa
$ kat hist -t 40 -H 250000000 -m 1000 -o kat_hist_abyss_k1000 -d SXPX_SEQ1K_100-3.fa
$ kat hist -t 40 -H 160000000 -m 1000 -o kat_hist_D3 B_DDN3_BASE-3.fa
```

To count 1,000-mers in the genome assemblies of three other Viridiplantae species (*C. reinhardtii*, *V. carteri*, and *A. thaliana*), the KAT software was employed. The following commands were issued:

```
$ kat hist -t 40 -H 150000000 -m 1000 -o Cr_kat_hist_k1000 Chlamydomonas_reinhardtii.fa
$ kat hist -t 40 -H 150000000 -m 1000 -o Vc_kat_hist_k1000 Volvox_carteri.fa
$ kat hist -t 40 -H 150000000 -m 1000 -o At_kat_hist_k1000 Arabidopsis_thaliana.fa
```

To determine the impact of repetitive k-mers on de Bruijn graph structure, the total set of 1,000-mers was first determined using the KAT software. The subsets of 1,000-mers with maximum allowed counts of 1, 2, 3, and 4, were then created using the Jellyfish software. The following commands were issued:

```
$ kat hist -t 40 -H 200000000 -m 1000 -o AB_k1000 -d BbB_ABruijn_v5.fa
$ jellyfish dump -U 1 -o B_A2DB_DUMP_U1.fasta AB_k1000-hash.jf1000
$ jellyfish dump -U 2 -o B_A2DB_DUMP_U2.fasta AB_k1000-hash.jf1000
$ jellyfish dump -U 3 -o B_A2DB_DUMP_U3.fasta AB_k1000-hash.jf1000
$ jellyfish dump -U 4 -o B_A2DB_DUMP_U4.fasta AB_k1000-hash.jf1000
```

The four subsets of 1,000-mers with variable maximum allowed counts were then re-assembled into de Bruijn graph contigs using BCALM2. The following commands were issued:

```
$ bcalm -nb-cores 20 -in B_A2DB_DUMP_U1.fasta -kmer-size 1000 -max-memory 52000 -abundance-min
1 -out B_A2DB_BASE_U1-1.fa
$ bcalm -nb-cores 20 -in B_A2DB_DUMP_U2.fasta -kmer-size 1000 -max-memory 52000 -abundance-min
1 -out B_A2DB_BASE_U2-1.fa
$ bcalm -nb-cores 20 -in B_A2DB_DUMP_U3.fasta -kmer-size 1000 -max-memory 52000 -abundance-min
1 -out B_A2DB_BASE_U3-1.fa
$ bcalm -nb-cores 20 -in B_A2DB_DUMP_U4.fasta -kmer-size 1000 -max-memory 52000 -abundance-min
1 -out B_A2DB_BASE_U4-1.fa
```

The de Bruijn graphs were re-constructed from the contigs and transformed into GFA format for visualization with Bandage using the ABySS toolkit. The following commands were issued:

```
$ abyss-overlap -v -j20 -m999 -k1000 --no-tred --dot --no-SS B_A2DB_BASE_U1-1.fa >
B_A2DB_BASE_U1-1.dot
$ abyss-filtergraph -v -k1000 --no-shim --gfa -g B_A2DB_BASE_U1-1.gfa B_A2DB_BASE_U1-1.dot >
tmp.path && rm tmp.path
$ abyss-overlap -v -j20 -m999 -k1000 --no-tred --dot --no-SS B_A2DB_BASE_U2-1.fa >
B_A2DB_BASE_U2-1.dot
```

```
$ abyss-filtergraph -v -k1000 --no-shim --gfa -g B_A2DB_BASE_U2-1.gfa B_A2DB_BASE_U2-1.dot >
tmp.path && rm tmp.path
$ abyss-overlap -v -j20 -m999 -k1000 --no-tred --dot --no-SS B_A2DB_BASE_U3-1.fa >
B_A2DB_BASE_U3-1.dot
$ abyss-filtergraph -v -k1000 --no-shim --gfa -g B_A2DB_BASE_U3-1.gfa B_A2DB_BASE_U3-1.dot >
tmp.path && rm tmp.path
$ abyss-overlap -v -j20 -m999 -k1000 --no-tred --dot --no-SS B_A2DB_BASE_U4-1.fa >
B_A2DB_BASE_U4-1.dot
$ abyss-filtergraph -v -k1000 --no-shim --gfa -g B_A2DB_BASE_U4-1.gfa B_A2DB_BASE_U4-1.dot >
tmp.path && rm tmp.path
```

Finally, the assembly statistics of each BCALM2 assembly, along with the original ABruijn assembly, were collected using the QUAST software. The following command was issued:

```
$ quast.py -o QUAST_v1 -t 1 -m 0 --plots-format png --contig-thresholds
0,1000,10000,100000,1000000 -l ABruijn,BCALM2-1,BCALM2-2,BCALM2-3,BCALM2-4 ../B_ABv5-1.fa
B_A2DB_BASE_U1-1.fa B_A2DB_BASE_U2-1.fa B_A2DB_BASE_U3-1.fa B_A2DB_BASE_U4-1.fa
```

A.2 Materials and Methods for Building the Version 2.0 Genome of *B. braunii*

Nearly all of the Illumina and PacBio data utilized in the assembly process described in this section are available from the JGI Genome Portal (https://genome.jgi.doe.gov/portal/) under JGI Project ID 1014520. The exception is the Illumina library LCHA, which is not currently available in public databases, but is available upon request.

A.2.1 Assembling the Illumina and PacBio Data

The base assembly for the Illumina data was generated from library SXPX. Prior to assembly, the library was filtered with Jellyfish and KAT to remove highly repetitive k-mers. To count and filter these k-mers, the following commands were issued:

```
$ jellyfish count -m 200 -s 30G -t 40 -L 1 -U 500 -o mer_counts_1-500.jf Library.SXPX.L.fq
Library.SXPX.R.fq
$ kat filter seq --stats -m 200 -t 40 -T 1 -o SXPX_KAT.L Library.SXPX.L.fq mer_counts_1-500.jf
$ kat filter seq --stats -m 200 -t 40 -T 1 -o SXPX_KAT.R Library.SXPX.R.fq mer_counts_1-500.jf
$ python fastqCombinePairedEnd.py SXPX_KAT.L.in.fq SXPX_KAT.R.in.fq
```

After filtering the read pairs separately, they needed to be matched together again into properly paired files for the assembly process. This was achieved with the following Python script (fastqCombinePairedEnd.py):

```
#!/usr/bin/env python
"""Resynchronize 2 fastq or fastq.gz files (R1 and R2) after they have been
trimmed and cleaned
WARNING! This program assumes that the fastq file uses EXACTLY four lines per
    sequence
Three output files are generated. The first two files contain the reads of the
    pairs that match and the third contains the solitary reads.
Usage:
    python fastqCombinePairedEnd.py input1 input2 separator
input1 = LEFT  fastq or fastq.gz file (R1)
input2 = RIGHT fastq or fastq.gz file (R2)
separator = character that separates the name of the read from the part that
    describes if it goes on the left or right, usually with characters '1' or
    '2'.  The separator is often a space, but could be another character. A
    space is used by default.
NOTE 09/22/16 DRB: Kindly taken from:
https://github.com/enormandeau/Scripts/blob/master/fastqCombinePairedEnd.py
"""

# Importing modules
import gzip
import sys

# Parsing user input
try:
    in1 = sys.argv[1]
    in2 = sys.argv[2]
except:
    print __doc__
    sys.exit(1)

try:
    separator = sys.argv[3]
except:
    separator = " "

# Defining classes
class Fastq(object):
    """Fastq object with name and sequence
    """

    def __init__(self, name, seq, name2, qual):
        self.name = name
        self.seq = seq
        self.name2 = name2
        self.qual = qual

    def getShortname(self, separator):
        self.temp = self.name.split(separator)
        del(self.temp[-1])
        return separator.join(self.temp)

    def write_to_file(self, handle):
```

```
        handle.write(self.name + "\n")
        handle.write(self.seq + "\n")
        handle.write(self.name2 + "\n")
        handle.write(self.qual + "\n")

# Defining functions
def myopen(infile, mode="r"):
    if infile.endswith(".gz"):
        return gzip.open(infile, mode=mode)
    else:
        return open(infile, mode=mode)

def fastq_parser(infile):
    """Takes a fastq file infile and returns a fastq object iterator
    """

    with myopen(infile) as f:
        while True:
            name = f.readline().strip()
            if not name:
                break

            seq = f.readline().strip()
            name2 = f.readline().strip()
            qual = f.readline().strip()
            yield Fastq(name, seq, name2, qual)

# Main
if __name__ == "__main__":
    seq1_dict = {}
    seq2_dict = {}
    seq1 = fastq_parser(in1)
    seq2 = fastq_parser(in2)
    s1_finished = False
    s2_finished = False

    if in1.endswith('.gz'):
        outSuffix='.fastq.gz'
    else:
        outSuffix='.fastq'

    with myopen(in1 + "_pairs_R1" + outSuffix, "w") as out1:
        with myopen(in2 + "_pairs_R2" + outSuffix, "w") as out2:
            with myopen(in1 + "_singles" + outSuffix, "w") as out3:
                while not (s1_finished and s2_finished):
                    try:
                        s1 = seq1.next()
                    except:
                        s1_finished = True
                    try:
                        s2 = seq2.next()
                    except:
                        s2_finished = True

                    # Add new sequences to hashes
                    if not s1_finished:
                        seq1_dict[s1.getShortname(separator)] = s1
                    if not s2_finished:
                        seq2_dict[s2.getShortname(separator)] = s2

                    if not s1_finished and s1.getShortname(separator) in seq2_dict:
```

```
            seq1_dict[s1.getShortname(separator)].write_to_file(out1)
            seq1_dict.pop(s1.getShortname(separator))
            seq2_dict[s1.getShortname(separator)].write_to_file(out2)
            seq2_dict.pop(s1.getShortname(separator))

        if not s2_finished and s2.getShortname(separator) in seq1_dict:
            seq2_dict[s2.getShortname(separator)].write_to_file(out2)
            seq2_dict.pop(s2.getShortname(separator))
            seq1_dict[s2.getShortname(separator)].write_to_file(out1)
            seq1_dict.pop(s2.getShortname(separator))

# Treat all unpaired reads
for r in seq1_dict.values():
    r.write_to_file(out3)

for r in seq2_dict.values():
    r.write_to_file(out3)
```

After filtering the library SXPX and re-matching the reads into proper pairs, the data were assembled with DISCOVAR de novo. After the primary assembly process, the output from DISCOVAR was further processed with tools from ABySS. The following commands were issued to assemble and process the data:

```
$ DiscovarDeNovo READS=SXPX_KAT.L.in.fq_pairs_R1.fastq,SXPX_KAT.R.in.fq_pairs_R2.fastq
OUT_DIR=DISCOVAR_v4 NUM_THREADS=40 MAX_MEM_GB=800
$ awk '/^>/{print (NR==1)?$0:"\n"$0;next}{printf "%s", $0}END{print ""}'
../DISCOVAR_v4/a.final/a.lines.fasta > B_DDN4_BASE-1.fa
$ awk '!/^>/ {next} {getline seq} $2 != "circular" {print $0 "\n" seq}' B_DDN4_BASE-1.fa >
B_DDN4_BASE-3.fa
$ awk '!/^>/ {next} {getline seq} $2 == "circular" {print $0 "\n" seq}' B_DDN4_BASE-1.fa >
B_DDN4_BASE-2.fa
$ abyss-overlap -v -k200 -m199 --no-SS B_DDN4_BASE-3.fa > B_DDN4_BASE-3.dot
$ abyss-filtergraph -v -k200 --no-SS --no-shim -t2000 -T1000 --assemble --gfa -g B_DDN4_BASE-
4.gfa B_DDN4_BASE-3.dot > B_DDN4_BASE-4.path
$ MergeContigs -v -k200 -o B_DDN4_BASE-5.fa -g B_DDN4_BASE-5.dot B_DDN4_BASE-3.fa B_DDN4_BASE-
4.gfa B_DDN4_BASE-4.path
$ abyss-filtergraph -v -k200 --no-shim --no-SS -l1000 --assemble --gfa -g B_DDN4_BASE-6.gfa
B_DDN4_BASE-5.dot > B_DDN4_BASE-6.path
$ MergeContigs -v -k200 -o B_DDN4_BASE-7.fa -g B_DDN4_BASE-7.dot B_DDN4_BASE-5.fa B_DDN4_BASE-
6.gfa B_DDN4_BASE-6.path
$ abyss-filtergraph -v -k200 --no-shim --no-SS --gfa -g B_DDN4_BASE-7.gfa B_DDN4_BASE-7.dot >
tmp.path && rm tmp.path
```

After the initial filtering with KAT and assembly with DISCOVAR, the contigs were scaffolded with all of the Illumina libraries (SXPX, NGNB, HOOW, LCHA) using HISAT2 to align the reads and BESST to perform the scaffolding. The following commands were issued to first align the reads and then generate scaffolds:

```
$ hisat2 --rf --no-spliced-alignment -I 3500 -X 6000 -p 20 -x ../B_DDN4_BASE-7.fa \
    -1 ../Reads/Library.HOOW.L.fq.gz \
    -2 ../Reads/Library.HOOW.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_HOOW \
    > HS2_HOOW.bam \
    && samtools index HS2_HOOW.bam
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 20 -x ../B_DDN4_BASE-7.fa \
    -1 ../Reads/Library_LCHA.L.fq \
    -2 ../Reads/Library_LCHA.R.fq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_LCHA \
    > HS2_LCHA.bam \
    && samtools index HS2_LCHA.bam
$ hisat2 --rf --no-spliced-alignment -I 1000 -X 3000 -p 20 -x ../B_DDN4_BASE-7.fa \
    -1 ../Reads/Library.NGNB.L.fq.gz \
    -2 ../Reads/Library.NGNB.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_NGNB \
    > HS2_NGNB.bam \
    && samtools index HS2_NGNB.bam
$ hisat2 --fr --no-spliced-alignment -I 300 -X 1000 -p 20 -x ../B_DDN4_BASE-7.fa \
    -1 ../Reads/SXPX_KAT.L.in.fq_pairs_R1.fastq \
    -2 ../Reads/SXPX_KAT.R.in.fq_pairs_R2.fastq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_SXPX \
    > HS2_SXPX.bam \
    && samtools index HS2_SXPX.bam
$ /scratch/user/dbrowne/Software/BESST/runBESST -c B_DDN4_BASE-7.fa \
    -f HS2_SXPX.bam HS2_NGNB.bam HS2_HOOW.bam HS2_LCHA.bam \
    -orientation fr rf rf fr -plots --separate_repeats --min_mapq 40 \
    --dfs_traversal -max_contig_overlap 200 -z_min 0.000001 -o ./D4_BESST_v1
```

After scaffolding, the assembly was polished with the Illumina data using HISAT2 again to align the reads and Pilon to process the data. The following commands were issued:

```
$ hisat2 --rf --no-spliced-alignment -I 3500 -X 6000 -p 20 -x ../Scaffolds-pass4.fa \
    -1 ../Reads/Library.HOOW.L.fq.gz \
    -2 ../Reads/Library.HOOW.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_HOOW \
    > HS2_HOOW.bam \
    && samtools index HS2_HOOW.bam
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 20 -x ../Scaffolds-pass4.fa \
    -1 ../Reads/Library_LCHA.L.fq \
    -2 ../Reads/Library_LCHA.R.fq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_LCHA \
    > HS2_LCHA.bam \
    && samtools index HS2_LCHA.bam
$ hisat2 --rf --no-spliced-alignment -I 1000 -X 3000 -p 20 -x ../Scaffolds-pass4.fa \
    -1 ../Reads/Library.NGNB.L.fq.gz \
    -2 ../Reads/Library.NGNB.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_NGNB \
    > HS2_NGNB.bam \
    && samtools index HS2_NGNB.bam
```

```
$ hisat2 --fr --no-spliced-alignment -I 300 -X 1000 -p 20 -x ../Scaffolds-pass4.fa \
    -1 ../Reads/SXPX_KAT.L.in.fq_pairs_R1.fastq \
    -2 ../Reads/SXPX_KAT.R.in.fq_pairs_R2.fastq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_SXPX \
    > HS2_SXPX.bam \
    && samtools index HS2_SXPX.bam
$ java -Xmx500G -jar $EBROOTPILON/pilon-1.20.jar --genome Scaffolds-pass4.fa \
    --frags Alignments/HS2_SXPX.bam --jumps Alignments/HS2_NGNB.bam \
    --jumps Alignments/HS2_HOOW.bam --jumps Alignments/HS2_LCHA.bam \
    --output D4BP --mingap 1 --K 25 --threads 40 --verbose --fix all,breaks
```

The PacBio data was assembled with ABruijn using a k-mer size of 14, which was shown to yield the best contiguity. The following command was issued:

```
$ abruijn.py -k 14 -t 20 -i 3 -o 3000 PACBIO_Reads_6kb_Min.fasta ABKO_k14 105
```

A.2.2 Merging the Illumina and PacBio Assemblies

To merge the Illumina and PacBio assemblies, unique 1,000-mers from each assembly were extracted and then combined using Jellyfish. The Jellyfish has was then dumped into a FASTA formatted file, which was then re-assembled with BCALM2. The following commands were issued:

```
$ jellyfish count -m 1000 -s 200000000 -U 1 -t 40 -o mer_counts_ABKO.jf REAPR_ABKO_k14-3.fa
$ jellyfish count -m 1000 -s 200000000 -U 1 -t 40 -o mer_counts_D4BP.jf REAPR_D4B+P.fa
$ jellyfish merge mer_counts_ABKO.jf mer_counts_D4BP.jf
$ jellyfish dump -o Bv2_k1000.fa mer_counts_merged.jf
$ bcalm -nb-cores 20 -in Bv2_k1000.fa -kmer-size 1000 -max-memory 52000 -abundance-min 1 -out
Bv2_BCALM
```

The BCALM2 output consists of sequences but does not contain any graph information. The de Bruijn graph can be reconstructed from the sequences by finding overlaps. The graph can then be processed to remove tips and pop bubbles. All of these functions can be achieved with tools in the ABySS toolkit. The following commands were issued:

```
# Re-build de Bruijn graph and transform to GFA spec
$ abyss-overlap -v -m999 -k1000 -j20 --no-tred --no-SS Bv2_BCALM.unitigs.fa >
Bv2_BCALM.unitigs.dot
```

```
$ abyss-filtergraph -v -k1000 --gfa --no-shim -g Bv2_BCALM.unitigs.gfa Bv2_BCALM.unitigs.dot >
tmp && rm tmp

# Filter tips <= 10 kb and assemble paths
$ abyss-filtergraph -v -k1000 --gfa --no-shim -t10000 --assemble -g Bv2_BCALM.tipless.gfa
Bv2_BCALM.unitigs.gfa > Bv2_BCALM.tipless.path

# Merge the contigs and re-write the graph
$ MergeContigs -v -k1000 -o Bv2_BCALM.tipless.fa -g Bv2_BCALM.tipless.dot Bv2_BCALM.unitigs.fa
Bv2_BCALM.tipless.gfa Bv2_BCALM.tipless.path

# Re-format Fasta headers
sed -i 's/LN:i://g' Bv2_BCALM.tipless.fa
sed -i 's/KC:i://g' Bv2_BCALM.tipless.fa

# Pop bubbles <= 10 kb and >= 90% identity
$ PopBubbles -v -j20 -k1000 -b10000 -p0.9 --no-SS --scaffold -g tmp Bv2_BCALM.tipless.fa
Bv2_BCALM.tipless.dot > Bv2_BCALM.popped.path && rm tmp

# Merge the contigs and re-write the graph (GRAPH WRITING FAILS)
$ MergeContigs -v -k1000 -o Bv2_BCALM.popped.fa -g Bv2_BCALM.popped.dot Bv2_BCALM.tipless.fa
Bv2_BCALM.tipless.dot Bv2_BCALM.popped.path

# Re-build de Bruijn graph (REWRITE GRAPH)
$ abyss-overlap -v -m999 -k1000 -j20 --no-tred --no-SS Bv2_BCALM.popped.fa >
Bv2_BCALM.popped.dot

# Filter contigs < 2000 bp
$ abyss-filtergraph -v -k1000 --gfa --no-shim -l2000 --assemble -g Bv2_BCALM.filter.gfa
Bv2_BCALM.popped.dot > Bv2_BCALM.filter.path

# Merge the contigs and transform graph to GFA spec
$ MergeContigs -v -k1000 -o Bv2_BCALM.contigs.fa -g Bv2_BCALM.contigs.dot Bv2_BCALM.popped.fa
Bv2_BCALM.filter.gfa Bv2_BCALM.filter.path
$ abyss-filtergraph -v -k1000--gfa --no-shim -g Bv2_BCALM.contigs.gfa Bv2_BCALM.contigs.dot >
tmp && rm tmp
```

A.2.3 Scaffolding, Gap Filling, and Polishing

The contigs were scaffolded using the four Illumina libraries (SXPX, NGNB, HOOW, LCHA) with the HISAT2 aligner and BESST scaffolder. The alignments and scaffolding were performed by issuing the following commands:

```
$ hisat2 --rf --no-spliced-alignment -I 3500 -X 6000 -p 20 -x ../Bv2_BCALM.contigs.fa \
    -1 ../Reads/Library.HOOW.L.fq.gz \
    -2 ../Reads/Library.HOOW.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_HOOW \
    > HS2_HOOW.bam \
    && samtools index HS2_HOOW.bam
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 20 -x ../Bv2_BCALM.contigs.fa \
    -1 ../Reads/Library_LCHA.L.fq \
```

```
    -2 ../Reads/Library_LCHA.R.fq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_LCHA \
    > HS2_LCHA.bam \
    && samtools index HS2_LCHA.bam
$ hisat2 --rf --no-spliced-alignment -I 1000 -X 3000 -p 20 -x ../Bv2_BCALM.contigs.fa \
    -1 ../Reads/Library.NGNB.L.fq.gz \
    -2 ../Reads/Library.NGNB.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_NGNB \
    > HS2_NGNB.bam \
    && samtools index HS2_NGNB.bam
$ hisat2 --fr --no-spliced-alignment -I 300 -X 1000 -p 20 -x ../Bv2_BCALM.contigs.fa \
    -1 ../Reads/SXPX_KAT.L.in.fq_pairs_R1.fastq \
    -2 ../Reads/SXPX_KAT.R.in.fq_pairs_R2.fastq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_SXPX \
    > HS2_SXPX.bam \
    && samtools index HS2_SXPX.bam
$ /scratch/user/dbrowne/Software/BESST/runBESST -c Bv2_BCALM.contigs.fa \
    -f HS2_SXPX.bam HS2_NGNB.bam HS2_HOOW.bam HS2_LCHA.bam \
    -orientation fr rf rf fr -plots --separate_repeats --min_mapq 40 \
    --dfs_traversal -max_contig_overlap 1000 -z_min 0.000001 -o ./Bv2_BESST_v1
```

REAPR analysis was applied to the scaffolds and errant linkages were broken. The follow commands were issued:

```
$ reapr facheck Scaffolds-pass4.fa Scaffolds-pass4.fa.facheck
$ hisat2-build Scaffolds-pass4.fa.facheck.fa Scaffolds-pass4.fa.facheck.fa
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 10 -x Scaffolds-pass4.fa.facheck.fa \
    -1 ../Reads/Library_LCHA.L.fq \
    -2 ../Reads/Library_LCHA.R.fq \
    | samtools view -@ 5 -Sb -f2 - \
    | samtools sort -@ 5 -O bam -T LCHA_v2_v1 \
    > LCHA_v2_v1.bam \
    && samtools index LCHA_v2_v1.bam
$ reapr pipeline Scaffolds-pass4.fa.facheck.fa LCHA_v2_v1.bam REAPR_v2_v1
```

Gap filling was performed on the broken scaffolds using the PacBio data with PBJelly. The PBJelly XML protocol was configured as follows:

```
<jellyProtocol>

<reference>/scratch/user/dbrowne/2017.03_MAR/2017.03.29_B_Genome_V2/Finishing_Round_1/03_PBJel
ly/Bv2_B1R2.fasta</reference>

<outputDir>/scratch/user/dbrowne/2017.03_MAR/2017.03.29_B_Genome_V2/Finishing_Round_1/03_PBJel
ly/Bv2_PBJelly</outputDir>
    <cluster>
        <command notes="For single node, multi-core machines" >${CMD} ${JOBNAME} 2> ${STDERR}
1> ${STDOUT} &amp;amp;</command>
        <nJobs>15</nJobs>
    </cluster>
```

```
    <blasr>-m 4 --hitPolicy allbest --nproc 1 --noSplitSubreads</blasr>
    <input
baseDir="/scratch/user/dbrowne/2017.03_MAR/2017.03.29_B_Genome_V2/Finishing_Round_1/03_PBJelly
/">
        <job>PACBIO_Reads_6kb_Min.fasta</job>
    </input>
</jellyProtocol>
```

To run the PBJelly pipeline, the following series of commands were issued:

```
Jelly.py setup Jelly_Protocol_v1.xml -x "--minGap=200 -i"
Jelly.py mapping Jelly_Protocol_v1.xml
Jelly.py support Jelly_Protocol_v1.xml -x "--spanOnly"
Jelly.py extraction Jelly_Protocol_v1.xml
Jelly.py assembly Jelly_Protocol_v1.xml -x "-w1000"
Jelly.py output Jelly_Protocol_v1.xml
```

REAPR was applied to the gap-filled assembly by issuing the following commands:

```
$ reapr facheck jelly.out.fasta jelly.out.fasta.facheck
$ hisat2-build jelly.out.fasta.facheck.fa jelly.out.fasta.facheck.fa
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 5 -x jelly.out.fasta.facheck.fa \
    -1 Reads/Library_LCHA.L.fq \
    -2 Reads/Library_LCHA.R.fq \
    | samtools view -@ 5 -Sb -f2 - \
    | samtools sort -@ 5 -O bam -T LCHA_v1 \
    > LCHA_v1.bam \
    && samtools index LCHA_v1.bam
$ reapr pipeline jelly.out.fasta.facheck.fa LCHA_v1.bam REAPR_v1
```

To polish the assembly with the Illumina data and Pilon, the following commands were issued:

```
$ hisat2 --rf --no-spliced-alignment -I 3500 -X 6000 -p 5 -x ../Bv2_BRJR.fa \
    -1 ../Reads/Library.HOOW.L.fq.gz \
    -2 ../Reads/Library.HOOW.R.fq.gz \
    | samtools view -@ 5 -Sb -f2 - \
    | samtools sort -@ 5 -O bam -T HS2_HOOW \
    > HS2_HOOW.bam \
    && samtools index HS2_HOOW.bam
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 5 -x ../Bv2_BRJR.fa \
    -1 ../Reads/Library_LCHA.L.fq \
    -2 ../Reads/Library_LCHA.R.fq \
    | samtools view -@ 5 -Sb -f2 - \
    | samtools sort -@ 5 -O bam -T HS2_LCHA \
    > HS2_LCHA.bam \
    && samtools index HS2_LCHA.bam
$ hisat2 --rf --no-spliced-alignment -I 1000 -X 3000 -p 5 -x ../Bv2_BRJR.fa \
    -1 ../Reads/Library.NGNB.L.fq.gz \
    -2 ../Reads/Library.NGNB.R.fq.gz \
    | samtools view -@ 5 -Sb -f2 - \
    | samtools sort -@ 5 -O bam -T HS2_NGNB \
    > HS2_NGNB.bam \
```

```
    && samtools index HS2_NGNB.bam
$ hisat2 --fr --no-spliced-alignment -I 300 -X 1000 -p 5 -x ../Bv2_BRJR.fa \
    -1 ../Reads/SXPX_KAT.L.in.fq_pairs_R1.fastq \
    -2 ../Reads/SXPX_KAT.R.in.fq_pairs_R2.fastq \
    | samtools view -@ 5 -Sb -f2 - \
    | samtools sort -@ 5 -O bam -T HS2_SXPX \
    > HS2_SXPX.bam \
    && samtools index HS2_SXPX.bam
$ java -Xmx500G -jar $EBROOTPILON/pilon-1.20.jar --genome Bv2_BRJR.fa \
    --frags Alignments/HS2_SXPX.bam --jumps Alignments/HS2_NGNB.bam \
    --jumps Alignments/HS2_HOOW.bam --jumps Alignments/HS2_LCHA.bam \
    --mingap 10 --mindepth 30 --minmq 40 --minqual 20 --nostrays \
    --output Bv2_BRJRP --K 25 --threads 40 --fix all
```

Finally, the assembly was polished with the PacBio data and Arrow by issuing the following commands:

```
$ blasr PACBIO_Reads_ALL.bam Bv2_BRJRP.final.fa --nproc 20 --bam --minSubreadLength 6000 --out
aligned_reads.bam --hitPolicy allbest
$ samtools sort -@ 20 aligned_reads.bam sorted_reads
$ pbindex sorted_reads.bam
$ samtools faidx Bv2_BRJRP.final.fa
$ arrow sorted_reads.bam -j 20 -r Bv2_BRJRP.final.fa -o variants.gff -o Bv2_BRJRPA.fa -o
Bv2_BRJRPA.fq
```

A.2.4 Quality Filtering and Re-scaffolding

To separate low- and high-quality scaffolds based on the FASTQ file provided by polishing with PacBio and Arrow, the following Python script was developed:

```
#!/usr/bin/env python

import sys

import numpy as np
from Bio import SeqIO

"""
Filter low-quality genome scaffolds after polishing with Arrow.

$ python Filter_Low_Qual.py polished_scaffolds.fq 30
"""

# Load input file into generator function for parsing
records = (r for r in SeqIO.parse(sys.argv[1], "fastq"))

# Open output files
scaffolds_HQ = open('Scaffolds_Q'+str(sys.argv[2])+'_HQ.fa', 'w')
scaffolds_LQ = open('Scaffolds_Q'+str(sys.argv[2])+'_LQ.fa', 'w')
quals = open('Quality_per_Scaffold_Q'+str(sys.argv[2])+'.tsv', 'w')
```

```
# Parse through records and write output accordingly
for record in records:
    scores = np.array([s for s in record.letter_annotations["phred_quality"]])
    quals.write('\t'.join([str(record.id),        str(scores.mean()),        str(scores.std()),
str(len(scores))])+'\n')
    if scores.mean() >= int(sys.argv[2]):
       scaffolds_HQ.write('>'+str(record.id)+'\n'+str(record.seq)+'\n')
    else:
       scaffolds_LQ.write('>'+str(record.id)+'\n'+str(record.seq)+'\n')

# Close output files
scaffolds_HQ.close()
scaffolds_LQ.close()
quals.close()
```

This script was called with the following command:

```
$ python Filter_Low_Qual.py Bv2_BRJRPA.fq 30
```

The low- and high-quality sequences were analyzed with QUAST by issuing the following command:

```
$ quast.py -o QUAST_Q30 -t 5 -l "Q30_HQ, Q30_LQ" -f -e \
    --contig-thresholds 0,10000,100000,1000000 \
    --gene-thresholds 0,1000,2000,3000 \
    --plots-format png \
    Scaffolds_Q30_HQ.fa \
    Scaffolds_Q30_LQ.fa
```

The low- and high-quality seqeuences were analyzed with REAPR and the Illumina library LCHA aligned with HISAT2 by issuing the following commands:

```
$ reapr facheck Scaffolds_Q30_HQ.fa Scaffolds_Q30_HQ.fa.facheck
$ hisat2-build Scaffolds_Q30_HQ.fa.facheck.fa Scaffolds_Q30_HQ.fa.facheck.fa
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 5 -x Scaffolds_Q30_HQ.fa.facheck.fa \
    -1 Reads/Library_LCHA.L.fq \
    -2 Reads/Library_LCHA.R.fq \
    | samtools view -@ 5 -Sb -f2 - \
    | samtools sort -@ 5 -O bam -T LCHA_Q30_HQ \
    > LCHA_Q30_HQ.bam \
    && samtools index LCHA_Q30_HQ.bam
$ reapr pipeline Scaffolds_Q30_HQ.fa.facheck.fa LCHA_Q30_HQ.bam REAPR_Q30_HQ
$ reapr facheck Scaffolds_Q30_LQ.fa Scaffolds_Q30_LQ.fa.facheck
$ hisat2-build Scaffolds_Q30_LQ.fa.facheck.fa Scaffolds_Q30_LQ.fa.facheck.fa
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 5 -x Scaffolds_Q30_LQ.fa.facheck.fa \
    -1 Reads/Library_LCHA.L.fq \
    -2 Reads/Library_LCHA.R.fq \
    | samtools view -@ 5 -Sb -f2 - \
    | samtools sort -@ 5 -O bam -T LCHA_Q30_LQ \
    > LCHA_Q30_LQ.bam \
```

```
    && samtools index LCHA_Q30_LQ.bam
$ reapr pipeline Scaffolds_Q30_LQ.fa.facheck.fa LCHA_Q30_LQ.bam REAPR_Q30_LQ$
```

The Illumina coverage profiles for each stage of the assembly were analyzed by aligning each Illumina library against each assembly with HISAT2. The following commands were issued:

```
$ hisat2 --rf --no-spliced-alignment -I 3500 -X 6000 -p 20 -x ../Bv2_BCALM.contigs.fa \
    -1 ../Reads/Library.HOOW.L.fq.gz \
    -2 ../Reads/Library.HOOW.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_HOOW \
    > HS2_HOOW.bam \
    && samtools index HS2_HOOW.bam
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 20 -x ../Bv2_BCALM.contigs.fa \
    -1 ../Reads/Library_LCHA.L.fq \
    -2 ../Reads/Library_LCHA.R.fq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_LCHA \
    > HS2_LCHA.bam \
    && samtools index HS2_LCHA.bam
$ hisat2 --rf --no-spliced-alignment -I 1000 -X 3000 -p 20 -x ../Bv2_BCALM.contigs.fa \
    -1 ../Reads/Library.NGNB.L.fq.gz \
    -2 ../Reads/Library.NGNB.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_NGNB \
    > HS2_NGNB.bam \
    && samtools index HS2_NGNB.bam
$ hisat2 --fr --no-spliced-alignment -I 300 -X 1000 -p 20 -x ../Bv2_BCALM.contigs.fa \
    -1 ../Reads/SXPX_KAT.L.in.fq_pairs_R1.fastq \
    -2 ../Reads/SXPX_KAT.R.in.fq_pairs_R2.fastq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_SXPX \
    > HS2_SXPX.bam \
    && samtools index HS2_SXPX.bam
$ hisat2 --rf --no-spliced-alignment -I 3500 -X 6000 -p 20 -x ../Bv2_BRJRPA.final.fa \
    -1 ../Reads/Library.HOOW.L.fq.gz \
    -2 ../Reads/Library.HOOW.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_HOOW \
    > HS2_HOOW.bam \
    && samtools index HS2_HOOW.bam
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 20 -x ../Bv2_BRJRPA.final.fa \
    -1 ../Reads/Library_LCHA.L.fq \
    -2 ../Reads/Library_LCHA.R.fq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_LCHA \
    > HS2_LCHA.bam \
    && samtools index HS2_LCHA.bam
$ hisat2 --rf --no-spliced-alignment -I 1000 -X 3000 -p 20 -x ../Bv2_BRJRPA.final.fa \
    -1 ../Reads/Library.NGNB.L.fq.gz \
    -2 ../Reads/Library.NGNB.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_NGNB \
    > HS2_NGNB.bam \
    && samtools index HS2_NGNB.bam
$ hisat2 --fr --no-spliced-alignment -I 300 -X 1000 -p 20 -x ../Bv2_BRJRPA.final.fa \
    -1 ../Reads/SXPX_KAT.L.in.fq_pairs_R1.fastq \
```

```
    -2 ../Reads/SXPX_KAT.R.in.fq_pairs_R2.fastq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_SXPX \
    > HS2_SXPX.bam \
    && samtools index HS2_SXPX.bam
$ hisat2 --rf --no-spliced-alignment -I 3500 -X 6000 -p 20 -x ../Scaffolds_Q30+_FILTERED.fa \
    -1 ../Reads/Library.HOOW.L.fq.gz \
    -2 ../Reads/Library.HOOW.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_HOOW \
    > HS2_HOOW.bam \
    && samtools index HS2_HOOW.bam
$ hisat2 --fr --no-spliced-alignment -I 5000 -X 30000 -p 20 -x ../Scaffolds_Q30+_FILTERED.fa \
    -1 ../Reads/Library_LCHA.L.fq \
    -2 ../Reads/Library_LCHA.R.fq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_LCHA \
    > HS2_LCHA.bam \
    && samtools index HS2_LCHA.bam
$ hisat2 --rf --no-spliced-alignment -I 1000 -X 3000 -p 20 -x ../Scaffolds_Q30+_FILTERED.fa \
    -1 ../Reads/Library.NGNB.L.fq.gz \
    -2 ../Reads/Library.NGNB.R.fq.gz \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_NGNB \
    > HS2_NGNB.bam \
    && samtools index HS2_NGNB.bam
$ hisat2 --fr --no-spliced-alignment -I 300 -X 1000 -p 20 -x ../Scaffolds_Q30+_FILTERED.fa \
    -1 ../Reads/SXPX_KAT.L.in.fq_pairs_R1.fastq \
    -2 ../Reads/SXPX_KAT.R.in.fq_pairs_R2.fastq \
    | samtools view -@ 10 -Sb - \
    | samtools sort -@ 10 -O bam -T HS2_SXPX \
    > HS2_SXPX.bam \
    && samtools index HS2_SXPX.bam
```

The final four alignments described aboved, aligned against the quality-filtered scaffolds, were used for re-scaffolding the assembly with BESST. The following command was issued:

```
/scratch/user/dbrowne/Software/BESST/runBESST -c Scaffolds_Q30+_FILTERED.fa \
    -f HS2_SXPX.bam HS2_NGNB.bam HS2_HOOW.bam HS2_LCHA.bam \
    -orientation fr rf rf fr -plots --min_mapq 40 -z_min 0 --separate_repeats \
    --dfs_traversal -max_contig_overlap 1000 -o ./Bv2_Q30+F_v1
```

The final breakage of the assembly, at regions with no fragment coverage (0x), was performed by Dr. Jerry Jenkins at the HudsonAlpha Institute for Biotechnology with a custom analytical pipeline. Briefly, the Illumina library LCHA was aligned against the assembly and fragmented at regions where there was no fragment coverage.

A.3 Materials and Methods for Application of Genome Annotation Methods

This section describes the tools utilized in the process of annotating the "Version 1.0" and "Version 2.0" genome assemblies to yield the v1.2 and v2.1 annotations, respectively.

A.3.1 Prediction of Protein-Coding Genes

In order to facilitate prediction of protein-coding genes, the JGI requested samples of RNA for sequencing (RNA-seq) with an Illumina HiSeq 2500. This presented an opportunity to conduct a meaningful RNA-seq experiment, which is presented in Section 4 of this document. For now, it is sufficient to state that approximately 1.1 billion fragments of RNA (270 bp targeted fragment size) were sequenced with 2x150 bp strand-specific, paired-end chemistry (2.2 billion total reads). Using this ultra-deep RNA-seq data, a genome-guided transcriptome assembly was made using PERTRAN (in-house program at JGI, unpublished). The RNA-seq data was also assembled at Texas A&M University using the Trinity *de novo* transcriptome assembler (*358, 374*). The PERTRAN and Trinity assemblies were consolidated with PASA, resulting in 55,930 assembled transcripts (*375*). Genomic loci were determined by alignments of proteins and transcripts using EXONERATE (*376*). Prior to alignment, repeats in the *B. braunii* genomic sequences were identified and soft-masked using RepeatMasker (*271*). The proteins used for alignment came from genomes available from the JGI database Phytozome (*276*), including *Volvox carteri*, *Coccomyxa subellipsoidea* C-169, *Micromonas pusilla* CCMP1545, *Micromonas* sp. RCC299, *Ostreococcus* sp. RCC809, *Ostreococcus lucimarinus*, *Chlamydomonas reinhardtii*, *Auxenochlorella protothecoides*, *Chlorella variabilis*, *Gonium pectorale*, *Helicosporidium* sp. ATCC 50920, *Monoraphidium neglectum*, *Ostreococcus tauri*, *Arabidopsis thaliana*, *Brachypodium distachyon*, *Sphagnum magellanicum*, and *Marchantia polymorpha*. Additional sequences used for alignment

came from the Marine Microbial Eukaryote Transcriptome Sequencing Project (MMETSP) and from the UniProt/SwissProt database (*377-379*).

In addition to determination of genomic loci by sequence alignments, gene models were generated with *ab initio* and homology-based methods, including FGENESH+ (*380*), GenomeScan (*381*), an in-house JGI homology-constrained ORF finder (using PASA assembly ORFs), and AUGUSTUS via BRAKER1 (*382*). The best-scored predictions for each locus were selected using multiple positive factors including sequence alignment support, and one negative factor: overlap with predicted repeat elements. The selected gene predictions were improved using PASA to add UTRs, splicing corrections, and alternative transcripts. Predicted *B. braunii* proteins were subjected to protein homology analysis against the aforementioned proteomes, to obtain C-scores and protein coverage. The C-score is defined as a ratio of protein BLASTP score to mutual best-hit BLASTP score. Protein coverage is defined as the highest percentage of protein aligned to the best of homologs.

B. braunii transcripts were selected based on C-score, protein coverage, EST coverage, and lack of CDS overlap with repeats. A transcript was selected if its C-score was larger than or equal to 0.5 and protein coverage was larger than or equal to 0.5, or if it had EST coverage, and if its CDS overlap with repeats was less than 20%. For gene models where CDS overlapped with repeats more that 20%, its C-score must be at least 0.9 and protein coverage at least 70% to be selected. The selected gene models were subject to protein family analysis with Pfam. For gene models where the protein is more than 30% in Pfam, transposable element domains were removed. Weak and incomplete gene models, with low protein coverage or without RNA-seq support, were removed from the set of predicted genes.

A.3.2 Functional Assignment to Proteins

With a predicted set of genes, transcripts, and proteins, functional classification methods can be applied to assign biological functions. Primarily, comparisons are made with databases of known sequences and functions are assigned by homology. Functional annotations can include such things as protein families (Pfam, PANTHER), Gene Ontology (GO) terms, Enzyme Commission (EC) numbers, Eukaryotic Orthologous Groups (KOG), Kyoto Encyclopedia of Genes and Genomes (KEGG) identifiers, etc. New classification systems will add further value to genome annotation efforts. Functions are assigned by JGI with the InterPro database (*383*) and the InterProScan classifier (*384*). The *B. braunii* v1.2 and v2.1 protein sets were given Pfam, PANTHER, and GO assignments with InterProScan using the default parameters. The assignment of EC numbers was done with the Ensemble Enzyme Prediction Pipeline (E2P2), a tool available from the Plant Metabolic Network (*385*). The KOG and KEGG assignments were made with RPS-BLAST and BLASTP, respectively. At Texas A&M University, InterProScan was independently applied to the v2.1 protein set, in an effort to assess the reproducibility of this annotation pipeline. This resulted in the creation of annotation set v2.1B, composed of the same proteins as v2.1, but with substantially different functional assignments.

A.3.3 Prediction of Repetitive Elements

Repeats were predicted in the "Version 1.0" and "Version 2.0" assemblies using RepeatModeler and RepeatMasker. The following commands were issued:

```
$ BuildDatabase -engine ncbi -name "BbB_V2" BbB_genome_V2.fa
$ RepeatModeler -database BbB_V2 -pa 20
$ RepeatMasker -pa 20 -e ncbi -gff -xsmall -no_is -dir REPEATS_v1 -lib BbB_V2.fa
BbB_genome_V2.fa

$ BuildDatabase -engine ncbi -name "BbB_V1" BbB_mainGenome.fasta
$ RepeatModeler -database BbB_V1 -pa 20
$ RepeatMasker -pa 20 -e ncbi -gff -xsmall -no_is -dir REPEATS_v1 -lib consensi.fa.classified
BbB_mainGenome.fasta
```

A.3.4 Prediction of DNA Methylation

DNA base modifications (i.e. methylation) were detected in the "Version 2.0" genome assembly using the kineticsTools software (https://github.com/PacificBiosciences/kineticsTools) to process BLASR (https://github.com/PacificBiosciences/blasr) alignments of the PacBio data. The following commands were issued:

```
$ blasr PACBIO_Reads_ALL.bam BbB_genome_V2.fa \
      --nproc 20 --bam --out aligned_reads_v2.bam --hitPolicy allbest \
      --minSubreadLength 4000 --maxScore -1000 --minAlnLength 1000 \
      --minPctSimilarity 80 --fastMaxInterval
$ samtools sort -@ 20 aligned_reads_v2.bam sorted_reads_v2
$ pbindex sorted_reads_v2.bam
$ samtools index BbB_genome_V2.fa
$ ipdSummary sorted_reads_v2.bam --reference BbB_genome_V2.fa \
      -v -j 20 --identify m6A,m4C --methylFraction \
      --gff methylation_v2.gff --csv methylation_v2.csv
```

APPENDIX B

SUPPLEMENTARY MATERIAL FOR COMPARATIVE GENOMICS OF VIRIDIPLANTAE

B.1 Materials and Methods for Functional Signatures in Genome Annotations

In order to determine the functional signatures of each genome by counting the frequencies of each annotation, the data was collected from the Phytozome database into a single directory. This was accomplished by issuing the following command:

```
$ cp /scratch/user/dbrowne/PhytozomeV12/*/annotation/*annotation_info* ./Functional_Data/
```

IMPORTANT NOTICE: there is an inconsistency in the naming of column 8 (KEGG/ec) in the annotation files. While 51 of the files utilize the convention "KEGG/ec", 13 of the files use only "ec". For the purposes of processing the data tables with a script, there needs to be uniformity among the annotation file headers. Therefore, prior to data processing, the column 8 headers were altered to all utilize "ec" as follows:

```
$ sed -i 's^KEGG/ec^ec^g' Function_Data/*
```

With the annotation files ready for analysis, a Python script was developed to parse, count, and process the annotation data. The script was run from the same directory that contained the “Functional_Data” sub-directory with all of the relevant annotations.

```
#!/usr/bin/env python
#

import numpy as np
import pandas as pd
import subprocess as sp

# Collect list of target files and parse into dictionary by species

targets = sp.Popen(['ls', './Functional_Data/'],
stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
targets = {x.split('_')[0]: x for x in targets}
```

```
# Load annotation files into dictionary of pandas dataframes

annotations = dict()
for k, v in targets.items():
    df = pd.read_csv('./Functional_Data/' + v, sep='\t', header=0, index_col=0)
    annotations[k] = df

# Load annotation data for Botryococcus braunii

bb_ec = pd.read_csv('./Bb_Data/EC_Term_Frequency.txt', sep=' ', names=['Bbraunii'],
index_col=0)
bb_go = pd.read_csv('./Bb_Data/GO_Term_Frequency.txt', sep=' ', names=['Bbraunii'],
index_col=0)
bb_ko = pd.read_csv('./Bb_Data/KO_Term_Frequency.txt', sep=' ', names=['Bbraunii'],
index_col=0)
bb_pf = pd.read_csv('./Bb_Data/PF_Term_Frequency.txt', sep=' ', names=['Bbraunii'],
index_col=0)

# Define functions to parse, count, and process data

def parse(df, term):
    loci = dict()
    for i, r in df.iterrows():
        if r[term] is np.NaN:
            continue
        t = set(r[term].split(','))
        try:
            loci[r['locusName']].update(t)
        except KeyError:
            loci[r['locusName']] = t
    return loci

def count(d):
    terms_list = list()
    terms_dict = dict()
    for v in d.values():
        terms_list += list(v)
    for t in terms_list:
        try:
            terms_dict[t] += 1
        except KeyError:
            terms_dict[t] = 1
    return terms_dict

def process(term, raw, log):
    d = dict()
    for k, v in annotations.items():
        locus_d = parse(v, term)
        count_d = count(locus_d)
        df = pd.DataFrame.from_dict(count_d, orient='index')
        df.columns = [k]
        d[k] = df
    m = {'ec': bb_ec, 'GO': bb_go, 'KO': bb_ko, 'Pfam': bb_pf}
    d['Bbraunii'] = m[term]

    data = pd.concat(d.values(), axis=1)
    data.fillna(0, inplace=True)
    data_raw = data.apply(lambda x: pd.to_numeric(x, errors='ignore', downcast='integer'))
    data_raw.to_csv(raw)
    data_log = data.apply(np.log1p)
    data_log.to_csv(log)
```

```
# Process annotations

ec = ('ec', './Functional_Signatures/EC_Analysis/EC_Term_Counts_Raw.csv',
'./Functional_Signatures/EC_Analysis/EC_Term_Counts_Log.csv')
go = ('GO', './Functional_Signatures/GO_Analysis/GO_Term_Counts_Raw.csv',
'./Functional_Signatures/GO_Analysis/GO_Term_Counts_Log.csv')
ko = ('KO', './Functional_Signatures/KO_Analysis/KO_Term_Counts_Raw.csv',
'./Functional_Signatures/KO_Analysis/KO_Term_Counts_Log.csv')
pf = ('Pfam', './Functional_Signatures/PF_Analysis/PF_Term_Counts_Raw.csv',
'./Functional_Signatures/PF_Analysis/PF_Term_Counts_Log.csv')

for term, raw, log in [ec, go, ko, pf]:
    process(term, raw, log)
```

With the annotation frequency tables, further analysis was conducted with additional Python scripts. The following script determined annotations that are found in every species of Viridiplantae:

```
#!/usr/bin/env python
#

import pandas as pd

ec_data = pd.read_csv('./EC_Analysis/EC_Term_Counts_Raw.csv', index_col=0, header=0)
go_data = pd.read_csv('./GO_Analysis/GO_Term_Counts_Raw.csv', index_col=0, header=0)
ko_data = pd.read_csv('./KO_Analysis/KO_Term_Counts_Raw.csv', index_col=0, header=0)
pf_data = pd.read_csv('./PF_Analysis/PF_Term_Counts_Raw.csv', index_col=0, header=0)

def process(data, out):
    out.write(','+','.join(list(data))+'\n')
    for i, row in data.iterrows():
        if 0 not in list(row):
            out.write(i+','+','.join([str(x) for x in row])+'\n')

with open('./EC_Analysis/EC_Term_Counts_Global_Viridiplantae.csv', 'w') as out:
    process(ec_data, out)
with open('./GO_Analysis/GO_Term_Counts_Global_Viridiplantae.csv', 'w') as out:
    process(go_data, out)
with open('./KO_Analysis/KO_Term_Counts_Global_Viridiplantae.csv', 'w') as out:
    process(ko_data, out)
with open('./PF_Analysis/PF_Term_Counts_Global_Viridiplantae.csv', 'w') as out:
    process(pf_data, out)
```

The following script determined annotations that are found in every species of Chlorophyta:

```
#!/usr/bin/env python
#

import pandas as pd
```

```
ec_data = pd.read_csv('./EC_Analysis/EC_Term_Counts_Raw.csv', index_col=0, header=0)
go_data = pd.read_csv('./GO_Analysis/GO_Term_Counts_Raw.csv', index_col=0, header=0)
ko_data = pd.read_csv('./KO_Analysis/KO_Term_Counts_Raw.csv', index_col=0, header=0)
pf_data = pd.read_csv('./PF_Analysis/PF_Term_Counts_Raw.csv', index_col=0, header=0)

chlorophyta = set(['Bbraunii', 'Creinhardtii', 'CsubellipsoideaC', 'Dsalina',
'MpusillaCCMP1545', 'MspRCC299', 'Olucimarinus', 'Vcarteri'])

def process(data, out):
    out.write(','+','.join(list(data))+'\n')
    for i, row in data.iterrows():
        if 0 not in [row[x] for x in chlorophyta]:
            out.write(i+','+','.join([str(x) for x in row])+'\n')

with open('./EC_Analysis/EC_Term_Counts_Global_Chlorophyta.csv', 'w') as out:
    process(ec_data, out)
with open('./GO_Analysis/GO_Term_Counts_Global_Chlorophyta.csv', 'w') as out:
    process(go_data, out)
with open('./KO_Analysis/KO_Term_Counts_Global_Chlorophyta.csv', 'w') as out:
    process(ko_data, out)
with open('./PF_Analysis/PF_Term_Counts_Global_Chlorophyta.csv', 'w') as out:
    process(pf_data, out)
```

The following script determined annotations that are found in every species of Embryophyta:

```
#!/usr/bin/env python
#

import pandas as pd

ec_data = pd.read_csv('./EC_Analysis/EC_Term_Counts_Raw.csv', index_col=0, header=0)
go_data = pd.read_csv('./GO_Analysis/GO_Term_Counts_Raw.csv', index_col=0, header=0)
ko_data = pd.read_csv('./KO_Analysis/KO_Term_Counts_Raw.csv', index_col=0, header=0)
pf_data = pd.read_csv('./PF_Analysis/PF_Term_Counts_Raw.csv', index_col=0, header=0)

chlorophyta = set(['Bbraunii', 'Creinhardtii', 'CsubellipsoideaC', 'Dsalina',
'MpusillaCCMP1545', 'MspRCC299', 'Olucimarinus', 'Vcarteri'])
embryophyta = sorted(set(ec_data.keys()) - set(chlorophyta))

def process(data, out):
    out.write(','+','.join(list(data))+'\n')
    for i, row in data.iterrows():
        if 0 not in [row[x] for x in embryophyta]:
            out.write(i+','+','.join([str(x) for x in row])+'\n')

with open('./EC_Analysis/EC_Term_Counts_Global_Embryophyta.csv', 'w') as out:
    process(ec_data, out)
with open('./GO_Analysis/GO_Term_Counts_Global_Embryophyta.csv', 'w') as out:
    process(go_data, out)
with open('./KO_Analysis/KO_Term_Counts_Global_Embryophyta.csv', 'w') as out:
    process(ko_data, out)
with open('./PF_Analysis/PF_Term_Counts_Global_Embryophyta.csv', 'w') as out:
    process(pf_data, out)
```

The following script determined the annotations that are missing only in *B. braunii* amongst the Viridiplantae:

```
#!/usr/bin/env python
#

import pandas as pd

ec_data = pd.read_csv('./EC_Analysis/EC_Term_Counts_Raw.csv', index_col=0, header=0)
go_data = pd.read_csv('./GO_Analysis/GO_Term_Counts_Raw.csv', index_col=0, header=0)
ko_data = pd.read_csv('./KO_Analysis/KO_Term_Counts_Raw.csv', index_col=0, header=0)
pf_data = pd.read_csv('./PF_Analysis/PF_Term_Counts_Raw.csv', index_col=0, header=0)

def process(data, out):
    out.write(',Bbraunii\n')
    for i, row in data.iterrows():
        nb = set(data.columns) - set(['Bbraunii'])
        v = set([row[x] for x in nb])
        if row['Bbraunii'] == 0 and 0 not in v:
            out.write(i+','+str(row['Bbraunii'])+'\n')

with open('./EC_Analysis/EC_Term_Counts_Missing_Viridiplantae.csv', 'w') as out:
    process(ec_data, out)
with open('./GO_Analysis/GO_Term_Counts_Missing_Viridiplantae.csv', 'w') as out:
    process(go_data, out)
with open('./KO_Analysis/KO_Term_Counts_Missing_Viridiplantae.csv', 'w') as out:
    process(ko_data, out)
with open('./PF_Analysis/PF_Term_Counts_Missing_Viridiplantae.csv', 'w') as out:
    process(pf_data, out)
```

The following script determined the annotations that are missing only in *B. braunii* amongst the Chlorophyta:

```
#!/usr/bin/env python
#

import pandas as pd

ec_data = pd.read_csv('./EC_Analysis/EC_Term_Counts_Raw.csv', index_col=0, header=0)
go_data = pd.read_csv('./GO_Analysis/GO_Term_Counts_Raw.csv', index_col=0, header=0)
ko_data = pd.read_csv('./KO_Analysis/KO_Term_Counts_Raw.csv', index_col=0, header=0)
pf_data = pd.read_csv('./PF_Analysis/PF_Term_Counts_Raw.csv', index_col=0, header=0)

chlorophyta = set(['Bbraunii', 'Creinhardtii', 'CsubellipsoideaC', 'Dsalina',
'MpusillaCCMP1545', 'MspRCC299', 'Olucimarinus', 'Vcarteri'])

def process(data, out):
    out.write(',Bbraunii\n')
    for i, row in data.iterrows():
        nb = chlorophyta - set(['Bbraunii'])
        v = set([row[x] for x in nb])
        if row['Bbraunii'] == 0 and 0 not in v:
            out.write(i+','+str(row['Bbraunii'])+'\n')

with open('./EC_Analysis/EC_Term_Counts_Missing_Chlorophyta.csv', 'w') as out:
```

```
    process(ec_data, out)
with open('./GO_Analysis/GO_Term_Counts_Missing_Chlorophyta.csv', 'w') as out:
    process(go_data, out)
with open('./KO_Analysis/KO_Term_Counts_Missing_Chlorophyta.csv', 'w') as out:
    process(ko_data, out)
with open('./PF_Analysis/PF_Term_Counts_Missing_Chlorophyta.csv', 'w') as out:
    process(pf_data, out)
```

The following script determined the annotations that are unique in *B. braunii* amongst the Viridiplantae:

```
#!/usr/bin/env python
#

import pandas as pd

ec_data = pd.read_csv('./EC_Analysis/EC_Term_Counts_Raw.csv', index_col=0, header=0)
go_data = pd.read_csv('./GO_Analysis/GO_Term_Counts_Raw.csv', index_col=0, header=0)
ko_data = pd.read_csv('./KO_Analysis/KO_Term_Counts_Raw.csv', index_col=0, header=0)
pf_data = pd.read_csv('./PF_Analysis/PF_Term_Counts_Raw.csv', index_col=0, header=0)

def process(data, out):
    out.write(',Bbraunii\n')
    for i, row in data.iterrows():
        if row['Bbraunii'] == sum(list(row)):
            out.write(i+','+str(row['Bbraunii'])+'\n')

with open('./EC_Analysis/EC_Term_Counts_Unique_Viridiplantae.csv', 'w') as out:
    process(ec_data, out)
with open('./GO_Analysis/GO_Term_Counts_Unique_Viridiplantae.csv', 'w') as out:
    process(go_data, out)
with open('./KO_Analysis/KO_Term_Counts_Unique_Viridiplantae.csv', 'w') as out:
    process(ko_data, out)
with open('./PF_Analysis/PF_Term_Counts_Unique_Viridiplantae.csv', 'w') as out:
    process(pf_data, out)
```

The following script determined the annotations that are unique in *B. braunii* amongst the Chlorophyta:

```
#!/usr/bin/env python
#

import pandas as pd

ec_data = pd.read_csv('./EC_Analysis/EC_Term_Counts_Raw.csv', index_col=0, header=0)
go_data = pd.read_csv('./GO_Analysis/GO_Term_Counts_Raw.csv', index_col=0, header=0)
ko_data = pd.read_csv('./KO_Analysis/KO_Term_Counts_Raw.csv', index_col=0, header=0)
pf_data = pd.read_csv('./PF_Analysis/PF_Term_Counts_Raw.csv', index_col=0, header=0)

chlorophyta = set(['Bbraunii', 'Creinhardtii', 'CsubellipsoideaC', 'Dsalina',
'MpusillaCCMP1545', 'MspRCC299', 'Olucimarinus', 'Vcarteri'])

def process(data, out):
```

```
    out.write(',Bbraunii\n')
    for i, row in data.iterrows():
        if row['Bbraunii'] > 0 and row['Bbraunii'] == sum([row[x] for x in chlorophyta]):
            out.write(i+','+str(row['Bbraunii'])+'\n')

with open('./EC_Analysis/EC_Term_Counts_Unique_Chlorophyta.csv', 'w') as out:
    process(ec_data, out)
with open('./GO_Analysis/GO_Term_Counts_Unique_Chlorophyta.csv', 'w') as out:
    process(go_data, out)
with open('./KO_Analysis/KO_Term_Counts_Unique_Chlorophyta.csv', 'w') as out:
    process(ko_data, out)
with open('./PF_Analysis/PF_Term_Counts_Unique_Chlorophyta.csv', 'w') as out:
    process(pf_data, out)
```

B.2 Materials and Methods for Evolution of Gene Organization in Genomes

In order to conduct further analyses of the genome sequences and gene structures, the genomes in the Phytozome database were collected into a single directory by issuing the following commands:

```
$ cp /scratch/user/dbrowne/PhytozomeV12/Acoerulea/assembly/Acoerulea_322_v3.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Acomosus/assembly/Acomosus_321_v3.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Ahalleri/assembly/Ahalleri_264_v1.fa.gz ./Genome_Data/
$ cp
/scratch/user/dbrowne/PhytozomeV12/Ahypochondriacus/assembly/Ahypochondriacus_315_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Alyrata/assembly/Alyrata_384_v1.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Athaliana/assembly/Athaliana_167_TAIR9.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Atrichopoda/assembly/Atrichopoda_291_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Bdistachyon/assembly/Bdistachyon_314_v3.0.fa.gz
./Genome_Data/
$ cp
/scratch/user/dbrowne/PhytozomeV12/Boleraceacapitata/assembly/Boleraceacapitata_446_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/BrapaFPsc/assembly/BrapaFPsc_277_v1.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Bstacei/assembly/Bstacei_316_v1.0.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Bstricta/assembly/Bstricta_278_v1.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Cclementina/assembly/Cclementina_182_v1.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Cgrandiflora/assembly/Cgrandiflora_266_v1.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Cpapaya/assembly/Cpapaya_113_r.Dec2008.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Creinhardtii/assembly/Creinhardtii_281_v5.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Crubella/assembly/Crubella_183_v1.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Csativus/assembly/Csativus_122_v1.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Csinensis/assembly/Csinensis_154_v1.fa.gz
./Genome_Data/
$ cp
/scratch/user/dbrowne/PhytozomeV12/CsubellipsoideaC169/assembly/CsubellipsoideaC_169_227_v2.0.
fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Dcarota/assembly/Dcarota_388_v2.0.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Dsalina/assembly/Dsalina_325_v1.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Egrandis/assembly/Egrandis_297_v2.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Esalsugineum/assembly/Esalsugineum_173_v1.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Fvesca/assembly/Fvesca_226_v1.1.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Gmax/assembly/Gmax_275_v2.0.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Graimondii/assembly/Graimondii_221_v2.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Kfedtschenkoi/assembly/Kfedtschenkoi_382_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Klaxiflora/assembly/Klaxiflora_309_v1.0.fa.gz
./Genome_Data/
```

```
$ cp
/scratch/user/dbrowne/PhytozomeV12/Lusitatissimum/assembly/Lusitatissimum_200_BGIv1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Macuminata/assembly/Macuminata_304_v1.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Mdomestica/assembly/Mdomestica_196_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Mesculenta/assembly/Mesculenta_305_v6.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Mguttatus/assembly/Mguttatus_256_v2.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Mpolymorpha/assembly/Mpolymorpha_320_v3.0.fa.gz
./Genome_Data/
$ cp
/scratch/user/dbrowne/PhytozomeV12/MpusillaCCMP1545/assembly/MpusillaCCMP1545_228_v3.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/MspRCC299/assembly/MspRCC299_229_v3.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Mtruncatula/assembly/Mtruncatula_285_Mt4.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Olucimarinus/assembly/Olucimarinus_231_v2.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Osativa/assembly/Osativa_323_v7.0.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Othomaeum/assembly/Othomaeum_386_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Phallii/assembly/Phallii_308_v2.0.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Ppatens/assembly/Ppatens_318_v3.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Ppersica/assembly/Ppersica_298_v2.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Ptrichocarpa/assembly/Ptrichocarpa_210_v3.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Pvirgatum/assembly/Pvirgatum_273_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Pvulgaris/assembly/Pvulgaris_442_v2.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Rcommunis/assembly/Rcommunis_119_TIGR.0.1.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Sbicolor/assembly/Sbicolor_313_v3.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Sfallax/assembly/Sfallax_310_v0.5.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Sitalica/assembly/Sitalica_312_v2.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Slycopersicum/assembly/Slycopersicum_390_v2.5.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Smoellendorffii/assembly/Smoellendorffii_91_v1.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Spolyrhiza/assembly/Spolyrhiza_290_v1.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Spurpurea/assembly/Spurpurea_289_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Stuberosum/assembly/Stuberosum_448_v4.03.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Sviridis/assembly/Sviridis_311_v1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Tcacao/assembly/Tcacao_233_CGDv1.0.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Tpratense/assembly/Tpratense_385_v2.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Vcarteri/assembly/Vcarteri_317_v2.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Vvinifera/assembly/Vvinifera_145_Genoscope.12X.fa.gz
./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/Zmarina/assembly/Zmarina_324_v2.2.fa.gz ./Genome_Data/
```

```
$ cp /scratch/user/dbrowne/PhytozomeV12/Zmays/assembly/Zmays_284_AGPv3.fa.gz ./Genome_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/ZmaysPH207/assembly/ZmaysPH207_443_v1.0.fa.gz
./Genome_Data/
```

Furthermore, the predicted protein sequences and gene exons were moved into separate directories for analysis as follows:

```
$ cp /scratch/user/dbrowne/PhytozomeV12/*/annotation/*protein_primary* ./Protein_Data/
$ cp /scratch/user/dbrowne/PhytozomeV12/*/annotation/*gene_exons* ./Predicted_Genes/
```

QUAST analysis was performed on the genome sequences to obtain contiguity statistics and other assembly descriptors by issuing the following command:

```
$ quast.py -o QUAST_v1 -t 20 -m 0 --plots-format png \
    --contig-thresholds 0,1000,10000,100000,1000000 \
    -l "Acoerulea, Acomosus, Ahalleri, Ahypochondriacus, Alyrata, Athaliana, Atrichopoda, \
    Bbraunii, Bdistachyon, Boleraceacapitata, BrapaFPsc, Bstacei, Bstricta, Cclementina, \
    Cgrandiflora, Cpapaya, Creinhardtii, Crubella, Csativus, Csinensis, CsubellipsoideaC, \
    Dcarota, Dsalina, Egrandis, Esalsugineum, Fvesca, Gmax, Graimondii, Kfedtschenkoi, \
    Klaxiflora, Lusitatissimum, Macuminata, Mdomestica, Mesculenta, Mguttatus, Mpolymorpha, \
    MpusillaCCMP1545, MspRCC299, Mtruncatula, Olucimarinus, Osativa, Othomaeum, Phallii, \
    Ppatens, Ppersica, Ptrichocarpa, Pvirgatum, Pvulgaris, Rcommunis, Sbicolor, Sfallax, \
    Sitalica, Slycopersicum, Smoellendorffii, Spolyrhiza, Spurpurea, Stuberosum, Sviridis, \
    Tcacao, Tpratense, Vcarteri, Vvinifera, Zmarina, Zmays, ZmaysPH207" \
    Genome_Data/Acoerulea_322_v3.fa \
    Genome_Data/Acomosus_321_v3.fa \
    Genome_Data/Ahalleri_264_v1.fa \
    Genome_Data/Ahypochondriacus_315_v1.0.fa \
    Genome_Data/Alyrata_384_v1.fa \
    Genome_Data/Athaliana_167_TAIR9.fa \
    Genome_Data/Atrichopoda_291_v1.0.fa \
    Genome_Data/Bbraunii_000_v2.fa \
    Genome_Data/Bdistachyon_314_v3.0.fa \
    Genome_Data/Boleraceacapitata_446_v1.0.fa \
    Genome_Data/BrapaFPsc_277_v1.fa \
    Genome_Data/Bstacei_316_v1.0.fa \
    Genome_Data/Bstricta_278_v1.fa \
    Genome_Data/Cclementina_182_v1.fa \
    Genome_Data/Cgrandiflora_266_v1.fa \
    Genome_Data/Cpapaya_113_r.Dec2008.fa \
    Genome_Data/Creinhardtii_281_v5.0.fa \
    Genome_Data/Crubella_183_v1.fa \
    Genome_Data/Csativus_122_v1.fa \
    Genome_Data/Csinensis_154_v1.fa \
    Genome_Data/CsubellipsoideaC_169_227_v2.0.fa \
    Genome_Data/Dcarota_388_v2.0.fa \
    Genome_Data/Dsalina_325_v1.fa \
    Genome_Data/Egrandis_297_v2.0.fa \
    Genome_Data/Esalsugineum_173_v1.fa \
    Genome_Data/Fvesca_226_v1.1.fa \
    Genome_Data/Gmax_275_v2.0.fa \
    Genome_Data/Graimondii_221_v2.0.fa \
    Genome_Data/Kfedtschenkoi_382_v1.0.fa \
```

```
    Genome_Data/Klaxiflora_309_v1.0.fa \
    Genome_Data/Lusitatissimum_200_BGIv1.0.fa \
    Genome_Data/Macuminata_304_v1.fa \
    Genome_Data/Mdomestica_196_v1.0.fa \
    Genome_Data/Mesculenta_305_v6.fa \
    Genome_Data/Mguttatus_256_v2.0.fa \
    Genome_Data/Mpolymorpha_320_v3.0.fa \
    Genome_Data/MpusillaCCMP1545_228_v3.0.fa \
    Genome_Data/MspRCC299_229_v3.0.fa \
    Genome_Data/Mtruncatula_285_Mt4.0.fa \
    Genome_Data/Olucimarinus_231_v2.0.fa \
    Genome_Data/Osativa_323_v7.0.fa \
    Genome_Data/Othomaeum_386_v1.0.fa \
    Genome_Data/Phallii_308_v2.0.fa \
    Genome_Data/Ppatens_318_v3.fa \
    Genome_Data/Ppersica_298_v2.0.fa \
    Genome_Data/Ptrichocarpa_210_v3.0.fa \
    Genome_Data/Pvirgatum_273_v1.0.fa \
    Genome_Data/Pvulgaris_442_v2.0.fa \
    Genome_Data/Rcommunis_119_TIGR.0.1.fa \
    Genome_Data/Sbicolor_313_v3.0.fa \
    Genome_Data/Sfallax_310_v0.5.fa \
    Genome_Data/Sitalica_312_v2.fa \
    Genome_Data/Slycopersicum_390_v2.5.fa \
    Genome_Data/Smoellendorffii_91_v1.fa \
    Genome_Data/Spolyrhiza_290_v1.fa \
    Genome_Data/Spurpurea_289_v1.0.fa \
    Genome_Data/Stuberosum_448_v4.03.fa \
    Genome_Data/Sviridis_311_v1.0.fa \
    Genome_Data/Tcacao_233_CGDv1.0.fa \
    Genome_Data/Tpratense_385_v2.fa \
    Genome_Data/Vcarteri_317_v2.fa \
    Genome_Data/Vvinifera_145_Genoscope.12X.fa \
    Genome_Data/Zmarina_324_v2.2.fa \
    Genome_Data/Zmays_284_AGPv3.fa \
    Genome_Data/ZmaysPH207_443_v1.0.fa
```

GenHub analyses were conducted by submitting jobs to the LSF batch scheduler using a Python script as follows:

```
#!/usr/bin/env python
#

import subprocess as sp

# Define template to run GenHub through LSF submission

command = """bsub -J Viridiplantae -L /bin/bash -W 20:00 -o Output/OUT_GENHUB_{0} \
-n 20 -R span[ptile=20] -R select[nxt] -R rusage[mem=2560] -M 2560 \
-cwd /scratch/user/dbrowne/2017.11_NOV/2017.11.25_Viridiplantae_GenHub \
ml restore GenHub; fidibus --workdir=./Results/ --numprocs=20 --local --label={0} \
--gdna=./Genomes/{1} \
--prot=./Proteins/{2} \
--gff3=./Annotations/{3} \
prep iloci breakdown stats cleanup"""

G = sorted(sp.Popen(['ls', './Genomes/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1])
```

```
P = sorted(sp.Popen(['ls', './Proteins/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1])
A = sorted(sp.Popen(['ls', './Annotations/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-
1])

targets = zip(G, P, A)
targets = {x[0].split('_')[0]: x for x in targets}

for k, v in targets.items():
    g, p, a = v
    sp.Popen(['mkdir', 'Results/' + k]).communicate()
    sp.Popen(command.format(k, g, p, a).split(' ')).communicate()
```

The following Python script was utilized to construct a boxplot of gene lengths from the GenHub data:

```
#!/usr/bin/env python

import subprocess as sp

import numpy as np
import pandas as pd

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns

# Load all relevant data for each species

targets = sp.Popen(['ls', '../Results/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
targets = {x: '../Results/' + x + '/' + x + '.pre-mrnas.tsv' for x in targets}

dataframes = dict()
for k, v in targets.items():
    df = pd.read_csv(v, sep='\t', header=0)
    dataframes[k] = df

quast = pd.read_csv('../../Viridiplantae_QUAST_Results_v1.csv', header=0, index_col=0)
genomes = {i: int(r['Total length (bp)'].translate(None, ', ')) for i, r in quast.iterrows()
if i in dataframes}
species = sorted(genomes, key=lambda x: genomes[x])

# Plot gene lengths from pre-mRNA data

gene_data = [dataframes[x]['Length'] for x in species]
x_ticks = [species[i] for i, x in enumerate(gene_data)]

sns.set(font_scale=3)
sns.set_style("ticks", {"ytick.minor.size": 4, "ytick.major.size": 7})
fig, ax = plt.subplots(figsize=(16, 8))
sns.boxplot(data=gene_data, palette="muted", ax=ax, whis=1.5, fliersize=3)

xtickNames = plt.setp(ax, xticklabels=x_ticks)
plt.setp(xtickNames, rotation=90, fontsize=14)

ax.set_yticks([i for i in range(0, 50000)[::500]], minor=True)
ax.yaxis.set_minor_locator(matplotlib.ticker.AutoMinorLocator(5))
```

```
plt.ylim(0, 50001)
plt.ylabel('Gene Length (bp)', fontsize=28)

fig.tight_layout()
plt.savefig('../BoxPlot_Figures/Gene_Lengths.png')
```

The following Python script was utilized to construct a boxplot of intergenic region lengths from the GenHub data:

```
#!/usr/bin/env python

import subprocess as sp

import numpy as np
import pandas as pd

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns

# Load all relevant data for each species

targets = sp.Popen(['ls', '../Results/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
targets = {x: '../Results/' + x + '/' + x + '.miloci.tsv' for x in targets}

dataframes = dict()
for k, v in targets.items():
    df = pd.read_csv(v, sep='\t', header=0)
    dataframes[k] = df

quast = pd.read_csv('../../Viridiplantae_QUAST_Results_v1.csv', header=0, index_col=0)
genomes = {i: int(r['Total length (bp)'].translate(None, ', ')) for i, r in quast.iterrows()
if i in dataframes}
species = sorted(genomes, key=lambda x: genomes[x])

# Plot iiLoci lengths from miLoci data

length_data = [dataframes[x][dataframes[x]['LocusClass'] == 'iiLocus']['Length'] for x in
species]
x_ticks = [species[i] for i, x in enumerate(length_data)]

sns.set(font_scale=3)
sns.set_style("ticks", {"ytick.minor.size": 4, "ytick.major.size": 7})
fig, ax = plt.subplots(figsize=(16, 8))
sns.boxplot(data=length_data, palette="muted", ax=ax, whis=1.5, fliersize=3)

xtickNames = plt.setp(ax, xticklabels=x_ticks)
plt.setp(xtickNames, rotation=90, fontsize=14)

ax.set_yticks([i for i in range(0, 50000)[::500]], minor=True)
ax.yaxis.set_minor_locator(matplotlib.ticker.AutoMinorLocator(5))
plt.ylim(0, 50001)
plt.ylabel('Intergenic Length (bp)', fontsize=28)

fig.tight_layout()
plt.savefig('../BoxPlot_Figures/Intergenic_Lengths.png')
```

The following Python script was utilized to construct a boxplot of exon lengths from the GenHub data:

```
#!/usr/bin/env python

import subprocess as sp

import numpy as np
import pandas as pd

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns

# Load all relevant data for each species

targets = sp.Popen(['ls', '../Results/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
targets = {x: '../Results/' + x + '/' + x + '.exons.tsv' for x in targets}

dataframes = dict()
for k, v in targets.items():
    df = pd.read_csv(v, sep='\t', header=0)
    dataframes[k] = df

quast = pd.read_csv('../../Viridiplantae_QUAST_Results_v1.csv', header=0, index_col=0)
genomes = {i: int(r['Total length (bp)'].translate(None, ', ')) for i, r in quast.iterrows()
if i in dataframes}
species = sorted(genomes, key=lambda x: genomes[x])

# Plot exon lengths from exon data

exon_data = [dataframes[x]['Length'] for x in species]
x_ticks = [species[i] for i, x in enumerate(exon_data)]

sns.set(font_scale=3)
sns.set_style("ticks", {"ytick.minor.size": 4, "ytick.major.size": 7})
fig, ax = plt.subplots(figsize=(16, 8))
sns.boxplot(data=exon_data, palette="muted", ax=ax, whis=1.5, fliersize=3)

xtickNames = plt.setp(ax, xticklabels=x_ticks)
plt.setp(xtickNames, rotation=90, fontsize=14)

ax.set_yticks([i for i in range(0, 5000)[::100]], minor=True)
ax.yaxis.set_minor_locator(matplotlib.ticker.AutoMinorLocator(5))
plt.ylim(0, 5001)
plt.ylabel('Exon Length (bp)', fontsize=28)

fig.tight_layout()
plt.savefig('../BoxPlot_Figures/Exon_Lengths.png')
```

The following Python script was utilized to construct a boxplot of intron lengths from the GenHub data:

```
#!/usr/bin/env python

import subprocess as sp

import numpy as np
import pandas as pd

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns

# Load all relevant data for each species

targets = sp.Popen(['ls', '../Results/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
targets = {x: '../Results/' + x + '/' + x + '.introns.tsv' for x in targets}

dataframes = dict()
for k, v in targets.items():
    df = pd.read_csv(v, sep='\t', header=0)
    dataframes[k] = df

quast = pd.read_csv('../../Viridiplantae_QUAST_Results_v1.csv', header=0, index_col=0)
genomes = {i: int(r['Total length (bp)'].translate(None, ', ')) for i, r in quast.iterrows()
if i in dataframes}
species = sorted(genomes, key=lambda x: genomes[x])

# Plot intron lengths from intron data

intron_data = [dataframes[x]['Length'] for x in species]
x_ticks = [species[i] for i, x in enumerate(intron_data)]

sns.set(font_scale=3)
sns.set_style("ticks", {"ytick.minor.size": 4, "ytick.major.size": 7})
fig, ax = plt.subplots(figsize=(16, 8))
sns.boxplot(data=intron_data, palette="muted", ax=ax, whis=1.5, fliersize=3)

xtickNames = plt.setp(ax, xticklabels=x_ticks)
plt.setp(xtickNames, rotation=90, fontsize=14)

ax.set_yticks([i for i in range(0, 5000)[::100]], minor=True)
ax.yaxis.set_minor_locator(matplotlib.ticker.AutoMinorLocator(5))
plt.ylim(0, 5001)
plt.ylabel('Intron Length (bp)', fontsize=28)

fig.tight_layout()
plt.savefig('../BoxPlot_Figures/Intron_Lengths.png')
```

The following Python script was utilized to construct a boxplot of 5’-UTR lengths from the GenHub data:

```
#!/usr/bin/env python

import subprocess as sp

import numpy as np
import pandas as pd
```

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns

# Load all relevant data for each species

targets = sp.Popen(['ls', '../Results/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
targets = {x: '../Results/' + x + '/' + x + '.pre-mrnas.tsv' for x in targets}

dataframes = dict()
for k, v in targets.items():
    df = pd.read_csv(v, sep='\t', header=0)
    dataframes[k] = df

quast = pd.read_csv('../../Viridiplantae_QUAST_Results_v1.csv', header=0, index_col=0)
genomes = {i: int(r['Total length (bp)'].translate(None, ', ')) for i, r in quast.iterrows()
if i in dataframes}
species = sorted(genomes, key=lambda x: genomes[x])

# Plot 3'-UTR lengths from pre-mRNA data

FpUTR_data = [dataframes[x][dataframes[x]['5pUTRlen'] > 0]['5pUTRlen'] for x in species]
x_ticks = [species[i] for i, x in enumerate(FpUTR_data)]

sns.set(font_scale=3)
sns.set_style("ticks", {"ytick.minor.size": 4, "ytick.major.size": 7})
fig, ax = plt.subplots(figsize=(16, 8))
sns.boxplot(data=FpUTR_data, palette="muted", ax=ax, whis=1.5, fliersize=3)

xtickNames = plt.setp(ax, xticklabels=x_ticks)
plt.setp(xtickNames, rotation=90, fontsize=14)

ax.set_yticks([i for i in range(0, 5000)[::100]], minor=True)
ax.yaxis.set_minor_locator(matplotlib.ticker.AutoMinorLocator(5))
plt.ylim(0, 5001)
plt.ylabel('5p-UTR Length (bp)', fontsize=28)

fig.tight_layout()
plt.savefig('../BoxPlot_Figures/5pUTR_Lengths.png')
```

The following Python script was utilized to construct a boxplot of CDS lengths from the GenHub data:

```
#!/usr/bin/env python

import subprocess as sp

import numpy as np
import pandas as pd

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load all relevant data for each species

targets = sp.Popen(['ls', '../Results/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
targets = {x: '../Results/' + x + '/' + x + '.cds.tsv' for x in targets}

dataframes = dict()
for k, v in targets.items():
    df = pd.read_csv(v, sep='\t', header=0)
    dataframes[k] = df

quast = pd.read_csv('../../Viridiplantae_QUAST_Results_v1.csv', header=0, index_col=0)
genomes = {i: int(r['Total length (bp)'].translate(None, ', ')) for i, r in quast.iterrows()
if i in dataframes}
species = sorted(genomes, key=lambda x: genomes[x])

# Plot CDS lengths from CDS data

cds_data = [dataframes[x]['Length'] for x in species]
x_ticks = [species[i] for i, x in enumerate(cds_data)]

sns.set(font_scale=3)
sns.set_style("ticks", {"ytick.minor.size": 4, "ytick.major.size": 7})
fig, ax = plt.subplots(figsize=(16, 8))
sns.boxplot(data=cds_data, palette="muted", ax=ax, whis=1.5, fliersize=3)

xtickNames = plt.setp(ax, xticklabels=x_ticks)
plt.setp(xtickNames, rotation=90, fontsize=14)

ax.set_yticks([i for i in range(0, 5000)[::100]], minor=True)
ax.yaxis.set_minor_locator(matplotlib.ticker.AutoMinorLocator(5))
plt.ylim(0, 5001)
plt.ylabel('CDS Length (bp)', fontsize=28)

fig.tight_layout()
plt.savefig('../BoxPlot_Figures/CDS_Lengths.png')
```

The following Python script was utilized to construct a boxplot of 3’-UTR lengths from the GenHub data:

```
#!/usr/bin/env python

import subprocess as sp

import numpy as np
import pandas as pd

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns

# Load all relevant data for each species

targets = sp.Popen(['ls', '../Results/'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
targets = {x: '../Results/' + x + '/' + x + '.pre-mrnas.tsv' for x in targets}

dataframes = dict()
```

```
for k, v in targets.items():
    df = pd.read_csv(v, sep='\t', header=0)
    dataframes[k] = df

quast = pd.read_csv('../../Viridiplantae_QUAST_Results_v1.csv', header=0, index_col=0)
genomes = {i: int(r['Total length (bp)'].translate(None, ', ')) for i, r in quast.iterrows()
if i in dataframes}
species = sorted(genomes, key=lambda x: genomes[x])

# Plot 3'-UTR lengths from pre-mRNA data

TpUTR_data = [dataframes[x][dataframes[x]['3pUTRlen'] > 0]['3pUTRlen'] for x in species]
x_ticks = [species[i] for i, x in enumerate(TpUTR_data)]

sns.set(font_scale=3)
sns.set_style("ticks", {"ytick.minor.size": 4, "ytick.major.size": 7})
fig, ax = plt.subplots(figsize=(16, 8))
sns.boxplot(data=TpUTR_data, palette="muted", ax=ax, whis=1.5, fliersize=3)

xtickNames = plt.setp(ax, xticklabels=x_ticks)
plt.setp(xtickNames, rotation=90, fontsize=14)

ax.set_yticks([i for i in range(0, 5000)[::100]], minor=True)
ax.yaxis.set_minor_locator(matplotlib.ticker.AutoMinorLocator(5))
plt.ylim(0, 5001)
plt.ylabel('3p-UTR Length (bp)', fontsize=28)

fig.tight_layout()
plt.savefig('../BoxPlot_Figures/3pUTR_Lengths.png')
```

B.3 Materials and Methods for Gene Evolution in Different Key Pathways

A custom Python script was developed, making use of the pandas and BioPython packages, in order to access the KEGG database and pull information about specific pathways. The pathway information was then utilized to parse the KEGG annotations associated with the Viridiplantae from the Phytozome database. The script was implemented as follows:

```
#!/usr/bin/env python
#

import pandas as pd
from Bio.KEGG import REST

# Load target KEGG pathway list and total raw count data

target = open('./Pathway_List.txt', 'r').read().split('\n')[:-1]
counts = pd.read_csv('../Functional_Signatures/KO_Analysis/KO_Term_Counts_Log.csv', header=0,
index_col=0)

# Function to parse orthology terms into dictionary structure

def extract_orthology(pw):
    D = dict()
    for line in pw:
        if not line.startswith(' ') and 'ORTHOLOGY' in D:
            break
        elif 'ORTHOLOGY' in D:
            D['ORTHOLOGY'].append(line.split())
        elif line.startswith('ORTHOLOGY'):
            D['ORTHOLOGY'] = [line.split()[1:]]
    d = ['_'.join(x[1:]) if len(x[1:]) > 1 else x[1] for x in D['ORTHOLOGY']]
    e = [x[0] for x in D['ORTHOLOGY']]
    return zip(e, d)

for t in target:
    p = [x for x in REST.kegg_get(t)]
    k = extract_orthology(p)
    n = '_'.join(p[1].split()[1:]) if len(p[1].split()) > 2 else p[1].split()[1]
    with open('./Log/' + n + '_Log.csv', 'w') as o:
        o.write(',' + ','.join(counts.columns) + '\n')
        i = set(counts.index.values)
        for e, d in k:
            if e in i:
                o.write(e + '_' + d.translate(None, ',') + ',' + ','.join([str(x) for x in
counts.loc[e]]) + '\n')
```

The list of KEGG pathways analyzed with the above script is as follows:

```
ko00190
ko00195
ko00196
ko00710
ko00020
```

ko00030
ko00630
ko00061
ko00230
ko00240
ko00430
ko00780
ko00130
ko00860
ko00900
ko00902
ko00909
ko00906
ko03020
ko03022
ko03040
ko03010
ko00970
ko03013
ko03015
ko03050
ko03030
ko03440
ko04075
ko04712
ko04626
ko04113
ko04210
ko04016
ko02010
ko04120
ko04110
ko00940
ko00941

APPENDIX C

SUPPLEMENTARY MATERIAL FOR DIEL CYCLES IN *BOTRYOCOCCUS BRAUNII*

C.1 Materials and Methods for Experimental Design and Biomass Collection

Strains and culture conditions were implemented as described previously (*127*). *B. braunii* race B (Showa) was grown in modified Chu-13 medium (*386*) using 13-W compact fluorescent 65 K lighting at a distance of 7.62 cm, which produced a light intensity of 280 μmol photons/m^2/s. Lighting was on a cycle of 12-h light/12-h dark at 22.5 °C. The cultures were continuously aerated by filter-sterilized air enriched with 2.5% CO_2. Fifty milliliters of culture were used to inoculate 750 mL of subsequent subcultures every 4 weeks. The culture medium contained KNO_3 (200 mg/L), $MgSO_4 \cdot 7H_2O$ (100 mg/L), $K_2HPO_4 \cdot 3H_2O$ (52 mg/L), $CaCI_2 \cdot 2H_2O$ (54 mg/L), FeNa EDTA (10 mg/L) and 5 mL of trace element solution per liter of culture medium. The trace element solution contained H_3BO_3 (572 mg/L), $MnSO_4 \cdot H_2O$ (308 mg/L), $ZnSO_4 \cdot 7H_2O$ (44 mg/L), $CuSO_4 \cdot 5H_2O$ (16mg/L), $Na_2MoO_2 \cdot 2H_2O$ (12 mg/L), $CoSO_4 \cdot 7H_2O$ (18 mg/L). The pH of the culture medium was adjusted to 7.2–7.5 with drops of 1 M H_2SO_4.

C.2 Materials and Methods for Analysis of Gene Expression

This section explains in detail the methods used for data processing and analysis of gene expression in the *B. braunii* v2.1 genome annotations.

C.2.1 RNA Extraction and Sequencing Results

The RNA sequencing data files are publicly available on the JGI Genome Portal database (Project ID 1139709). The file names are uninterpretable without access to the metadata worksheet describing the sample origins. In order to make the files intelligible, a Python script was developed to create a Shell script to rename all of the files, as follows:

```
#!/usr/bin/env python

data = open("Data_Renaming_Table.txt", "r").read().split('\n')
data = [x.split(' ') for x in data]

f = ['.chaff.tar', '.filter-RNA.fastq.gz', '.filter_cmd-RNA.sh', '.filtered-methods.txt',
'.filtered-report.txt']

with open('Renaming_Script.sh', 'w') as out:
    for n in data:
        for x in f:
            out.write(' '.join(['mv', n[0]+x, n[1]+x])+'\n')
```

The data renaming table utilized in the above script is as follows:

```
11457.4.207909.TGTGCGT-AACGCAC BbB_D21_0500_R1
11457.4.207909.ACCATCC-TGGATGG BbB_D21_0500_R2
11457.8.207939.GCTACGT-AACGTAG BbB_D21_0500_R3
11457.8.207939.CGCTTAA-GTTAAGC BbB_D21_1100_R1
11463.1.207962.TCCGAGT-AACTCGG BbB_D21_1100_R2
11463.1.207962.AGTCTCA-GTGAGAC BbB_D21_1100_R3
11457.4.207909.CCTCAGT-AACTGAG BbB_D21_1700_R1
11463.2.207969.ACGGTCT-AAGACCG BbB_D21_1700_R3
11463.2.207969.GTAACGA-GTCGTTA BbB_D21_2300_R1
11463.2.207969.ATTGAGC-GGCTCAA BbB_D21_2300_R3
11463.3.207976.GTGAGCT-AAGCTCA BbB_D22_0500_R1
11457.7.207933.CAATCGA-GTCGATT BbB_D22_0500_R2
11457.4.207909.CCTTCCT-AAGGAAG BbB_D22_0500_R3
11463.3.207976.TGACTGA-GTCAGTC BbB_D22_1100_R1
11457.4.207909.GTCTCCT-AAGGAGA BbB_D22_1100_R2
11463.3.207976.ACGATGA-GTCATCG BbB_D22_1100_R3
11457.4.207909.TACGCCT-AAGGCGT BbB_D22_1700_R1
11463.2.207969.AGTAGTC-GGACTAC BbB_D22_1700_R2
11457.8.207939.AGAGCCT-AAGGCTC BbB_D22_1700_R3
11463.2.207969.TCTCTTC-GGAAGAG BbB_D22_2300_R1
```

```
11457.8.207939.GAGGACT-AAGTCCT BbB_D22_2300_R2
11463.2.207969.AGAATGC-GGCATTC BbB_D22_2300_R3
11457.8.207939.GCTGGAT-AATCCAG BbB_D23_0500_R1
11457.7.207933.AGCAAGC-TGCTTGC BbB_D23_0500_R2
11457.8.207939.CGACCAT-AATGGTC BbB_D23_0500_R3
11463.3.207976.TCGGTTA-GTAACCG BbB_D23_1100_R1
11463.1.207962.CACGTTG-ACAACGT BbB_D23_1100_R2
11463.3.207976.GTTCAAC-GGTTGAA BbB_D23_1100_R3
11463.1.207962.CAGAGTG-ACACTCT BbB_D23_1700_R1
11457.7.207933.GGATACC-TGGTATC BbB_D23_1700_R2
11463.1.207962.ACCTCTG-ACAGAGG BbB_D23_1700_R3
11457.7.207933.TGGATCA-GTGATCC BbB_D23_2300_R1
11463.1.207962.CTTGCTG-ACAGCAA BbB_D23_2300_R2
11457.7.207933.GCCATAA-GTTATGG BbB_D23_2300_R3
```

Once all of the files were renamed, the interleaved reads in the FASTQ files were separated into separate pairs of FASTQ files for the left-end and right-end reads. This was achieved using the following Python script:

```
#!/usr/bin/env python

import sys
import gzip
import subprocess as sp

# usage: python FastQ_Unweave.py interleaved_reads.fq.gz output_reads_L output_reads_R

input_reads = gzip.open(sys.argv[1], 'rb')
output_L = gzip.open(sys.argv[2]+'.fq.gz', 'wb')
output_R = gzip.open(sys.argv[3]+'.fq.gz', 'wb')

def pair_reader(reads):
    read_L = [reads.readline() for _ in range(4)]
    read_R = [reads.readline() for _ in range(4)]
    return (read_L, read_R)

def pair_counter(reads):
    p1 = sp.Popen(['zcat', sys.argv[1]], stdout=sp.PIPE)
    p2 = sp.Popen(['wc', '-l'], stdin=p1.stdout, stdout=sp.PIPE)
    pairs = int(p2.communicate()[0]) / 8
    return range(pairs)

for i in pair_counter(sys.argv[1]):
        read_L, read_R = pair_reader(input_reads)
    output_L.write(''.join(read_L))
    output_R.write(''.join(read_R))
```

C.2.2 Quality Control of Biological Replicates

The “Alignment v1” data were obtained by issuing the following set of commands:

```
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
```

```
    -1 ../Separated_Pairs/BbB_D21_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
```

```
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R1_R.fq.gz \
```

```
    | samtools view -b > BAM_Files/D23_0500_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
```

```
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R3.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R1.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R2.bam"
bsub -J RNAseq_Alignment_v1 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v1" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R3.bam"
```

The “Alignment v2” data were obtained by issuing the following set of commands:

```
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
```

```
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R2 \
```

```
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R3_L.fq.gz \
```

```
    -2 ../Separated_Pairs/BbB_D22_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
```

```
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R3.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R1.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R2.bam"
bsub -J RNAseq_Alignment_v2 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v2" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R3.bam"
```

The “Alignment v3” data were obtained by issuing the following set of commands:

```
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R1.bam"
```

```
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
```

```
    -1 ../Separated_Pairs/BbB_D22_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
```

```
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R2_R.fq.gz \
```

```
    | samtools view -b > BAM_Files/D23_1700_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R3.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R1.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R2.bam"
bsub -J RNAseq_Alignment_v3 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v3" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R3.bam"
```

The “Alignment v4” data were obtained by issuing the following set of commands:

```
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R3_R.fq.gz \
```

```
    | samtools view -b > BAM_Files/D21_0500_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D21_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
```

```
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R3 \
```

```
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D22_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R1_L.fq.gz \
```

```
    -2 ../Separated_Pairs/BbB_D23_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R3.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R1.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R2.bam"
bsub -J RNAseq_Alignment_v4 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v4" \
```

```
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --tmo -p 20 -x Scaffolds-pass4.broken.0x.fasta \
    -1 ../Separated_Pairs/BbB_D23_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R3.bam"
```

The “Alignment v5” data were obtained by issuing the following set of commands:

```
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_0500_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1100_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
```

```
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_1700_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D21_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D21_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D21_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D21_2300_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_0500_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R1.bam"
```

```
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1100_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_1700_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D22_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D22_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
```

```
    -1 ../Separated_Pairs/BbB_D22_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D22_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D22_2300_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_0500_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_0500_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_0500_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_0500_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_0500_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_0500_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_1100_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_1100_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_1100_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_1100_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1100_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1100_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_1700_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
```

```
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_1700_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_1700_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_1700_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_1700_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_1700_R3.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R1 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_2300_R1_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R1_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R1.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R2 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_2300_R2_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R2_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R2.bam"
bsub -J RNAseq_Alignment_v5 -L /bin/bash -W 10:00 -o Output/OUT_D23_2300_R3 \
    -n 20 -R "span[ptile=20] select[nxt] rusage[mem=2700]" -M 2700 \
    -cwd "/scratch/user/dbrowne/2017.11_NOV/2017.11.09_RNAseq_Data_Analysis/Alignment_v5" \
    "ml HISAT2/2.0.5-intel-2015B-Python-2.7.10; ml SAMtools/1.3-intel-2015B; \
    hisat2 --rna-strandness RF --no-spliced-alignment -p 20 -x Bbrauniiv2.1.primaryTrs.fa \
    -1 ../Separated_Pairs/BbB_D23_2300_R3_L.fq.gz \
    -2 ../Separated_Pairs/BbB_D23_2300_R3_R.fq.gz \
    | samtools view -b > BAM_Files/D23_2300_R3.bam"
```

Quantification and sample analysis of all the libraries prior to QC filtering was achieved by submitting the following job script:

```
#BSUB -J QUANTIFY-QC_Kv1A -L /bin/bash -W 5:00
#BSUB -n 20 -R "span[ptile=20] rusage[mem=2700] select[nxt]" -M 2700
#BSUB -o OUT_QUANTIFY-QC_Kv1A
##
cd /scratch/user/dbrowne/2018.02_FEB/2018.02.01_B_RNAseq_Analysis/KALLISTO_v1/Kv1A
#
ml Trinity/2.5.1-GCCcore-6.3.0-Perl-5.24.0
ml SAMtools/1.6-GCCcore-6.3.0
ml kallisto/0.43.1-intel-2017A
ml R/3.4.2-intel-2017A-Python-2.7.12-default-mt
export R_LIBS=/scratch/user/dbrowne/Software/R_libs
#
ln -s ../../Bbrauniishowav2.1.primaryTrs.fa
$TRINITY_HOME/util/align_and_estimate_abundance.pl \
    --transcripts Bbrauniishowav2.1.primaryTrs.fa \
```

```
    --seqType fq --prep_reference \
    --samples_file ../../Sample_List_v1.txt \
    --thread_count 20 --est_method kallisto \
    --kallisto_add_opts "-t 20 --rf-stranded --bias"
#
ls -1 */abundance.tsv | awk '{print $9}' > Kv1A_quant_files.txt
$TRINITY_HOME/util/abundance_estimates_to_matrix.pl \
    --est_method kallisto --quant_files Kv1A_quant_files.txt \
    --name_sample_by_basedir --out_prefix Kv1A --gene_trans_map none
#
$TRINITY_HOME/Analysis/DifferentialExpression/PtR \
    --matrix Kv1A.isoform.counts.matrix \
    --samples ../../Sample_List_v1.txt --CPM --log2 \
    --compare_replicates
#
$TRINITY_HOME/Analysis/DifferentialExpression/PtR \
    --matrix Kv1A.isoform.counts.matrix \
    --samples ../../Sample_List_v1.txt \
    --log2 --CPM --sample_cor_matrix
#
$TRINITY_HOME/Analysis/DifferentialExpression/PtR \
    --matrix Kv1A.isoform.counts.matrix \
    --samples ../../Sample_List_v1.txt \
    --log2 --CPM --prin_comp 3 --center_rows
```

Quantification and sample analysis of the remaining libraries after QC filtering was achieved by submitting the following job script:

```
#BSUB -J QUANTIFY-QC_Kv2A -L /bin/bash -W 5:00
#BSUB -n 20 -R "span[ptile=20] rusage[mem=2700] select[nxt]" -M 2700
#BSUB -o OUT_QUANTIFY-QC_Kv2A
##
cd /scratch/user/dbrowne/2018.02_FEB/2018.02.01_B_RNAseq_Analysis/KALLISTO_v2/Kv2A
#
ml Trinity/2.5.1-GCCcore-6.3.0-Perl-5.24.0
ml SAMtools/1.6-GCCcore-6.3.0
ml kallisto/0.43.1-intel-2017A
ml R/3.4.2-intel-2017A-Python-2.7.12-default-mt
export R_LIBS=/scratch/user/dbrowne/Software/R_libs
#
ln -s ../../Bbrauniishowav2.1.primaryTrs.fa
$TRINITY_HOME/util/align_and_estimate_abundance.pl \
    --transcripts Bbrauniishowav2.1.primaryTrs.fa \
    --seqType fq --prep_reference \
    --samples_file ../../Sample_List_v2.txt \
    --thread_count 20 --est_method kallisto \
    --kallisto_add_opts "-t 20 --rf-stranded --bias"
#
ls -1 */abundance.tsv | awk '{print $9}' > Kv2A_quant_files.txt
$TRINITY_HOME/util/abundance_estimates_to_matrix.pl \
    --est_method kallisto --quant_files Kv2A_quant_files.txt \
    --name_sample_by_basedir --out_prefix Kv2A --gene_trans_map none
#
$TRINITY_HOME/Analysis/DifferentialExpression/PtR \
    --matrix Kv2A.isoform.counts.matrix \
    --samples ../../Sample_List_v2.txt --CPM --log2 \
    --compare_replicates
```

```
#
$TRINITY_HOME/Analysis/DifferentialExpression/PtR \
    --matrix Kv2A.isoform.counts.matrix \
    --samples ../../Sample_List_v2.txt \
    --log2 --CPM --sample_cor_matrix
#
$TRINITY_HOME/Analysis/DifferentialExpression/PtR \
    --matrix Kv2A.isoform.counts.matrix \
    --samples ../../Sample_List_v2.txt \
    --log2 --CPM --prin_comp 3 --center_rows
```

C.2.3 Differential Gene Expression Analysis

After QC filtering, the Trinity differential gene expression analysis was performed with the quantification data by issuing the following job script:

```
#BSUB -J DGE_ANALYSIS_v1 -L /bin/bash -W 5:00
#BSUB -n 10 -R "span[ptile=10] rusage[mem=2700] select[nxt]" -M 2700
#BSUB -o OUT_DGE_ANALYSIS_v1
##
cd /scratch/user/dbrowne/2018.02_FEB/2018.02.01_B_RNAseq_Analysis/
#
ml Trinity/2.5.1-GCCcore-6.3.0-Perl-5.24.0
ml SAMtools/1.6-GCCcore-6.3.0
ml kallisto/0.43.1-intel-2017A
ml R/3.4.2-intel-2017A-Python-2.7.12-default-mt
export R_LIBS=/scratch/user/dbrowne/Software/R_libs
#
$TRINITY_HOME/Analysis/DifferentialExpression/run_DE_analysis.pl \
          --matrix KALLISTO_v2/Kv2A/Kv2A.isoform.counts.matrix \
          --min_reps_min_cpm 5,1 --method voom \
          --output DGE_ANALYSIS_v1 \
          --samples_file Sample_List_v2.txt
```

In order to test the effects of different p-value (P) and fold-change (C) thresholds, the following Python script was developed:

```
#!/usr/bin/env python

import os
import itertools as it
import subprocess as sp

cmd = """../analyze_diff_expr.pl \
    --matrix ../../KALLISTO_v2/Kv2A/Kv2A.isoform.TMM.EXPR.matrix \
    --samples ../../Sample_List_v2.txt \
    --output DGE_v1 -P {} -C {} \
    --max_genes_clust 20000"""

P = ['1e-' + str(i) for i in range(2, 10)]
```

```
C = [str(i) for i in range(1, 5)]
F = sp.Popen(['ls'], stdout=sp.PIPE).communicate()[0].split('\n')[:-1]
F = [f for f in F if 'Kv2A' in f]

for p, c in it.product(P, C):
    os.mkdir('./P' + p + '_C' + c)
    os.chdir('./P' + p + '_C' + c)
    for f in F:
        sp.Popen(['cp', '../' + f, './']).communicate()
    sp.Popen(cmd.format(p, c).split()).communicate()
    os.chdir('../')
```

To run the experimental script described above, the following job script was issued:

```
#BSUB -J PC_TESTING_v1 -L /bin/bash -W 5:00
#BSUB -n 20 -R "span[ptile=20] rusage[mem=2700] select[nxt]" -M 2700
#BSUB -o OUT_PC_TESTING_v1
##
cd /scratch/user/dbrowne/2018.02_FEB/2018.02.01_B_RNAseq_Analysis/DGE_ANALYSIS_v1
#
ml Trinity/2.5.1-GCCcore-6.3.0-Perl-5.24.0
ml SAMtools/1.6-GCCcore-6.3.0
ml kallisto/0.43.1-intel-2017A
ml R/3.4.2-intel-2017A-Python-2.7.12-default-mt
export R_LIBS=/scratch/user/dbrowne/Software/R_libs
#
python PC_Testing_v1.py
```

C.2.4 Coexpression of Genes and Functions

A custom Python script was developed to utilize the cluster cutting tool from Trinity and analyze the effects of cutting clusters at different percents of tree height:

```
#!/usr/bin/env python

import os
import itertools as it
import subprocess as sp

cmd = "../define_clusters_by_cutting_tree.pl --Ptree {} -R  {}"

Pt = [str(i) for i in range(20, 81, 10)]
Rd = ["../DGE_v1.P1e-2C0.matrix.RData",
    "../DGE_v1.P1e-2C1.matrix.RData",
    "../DGE_v1.P1e-2C2.matrix.RData",
    "../DGE_v1.P1e-2C3.matrix.RData",
    "../DGE_v1.P1e-2C4.matrix.RData"]

for pt, rd in it.product(Pt, Rd):
    os.mkdir('./Pt' + pt + '_' + rd.split('.')[-3])
    os.chdir('./Pt' + pt + '_' + rd.split('.')[-3])
    sp.Popen(cmd.format(pt, rd).split()).communicate()
```

```
    os.chdir('../')
```

The following job script was issued to perform the cluster cutting analysis with the script described above:

```
#BSUB -J Cluster_Cut_v1 -L /bin/bash -W 5:00
#BSUB -n 5 -R "span[ptile=5] rusage[mem=2700] select[nxt]" -M 2700
#BSUB -o OUT_CLUSTER_CUT_v1
##
cd /scratch/user/dbrowne/2018.02_FEB/2018.02.01_B_RNAseq_Analysis/Cluster_Cutting
#
ml Trinity/2.5.1-GCCcore-6.3.0-Perl-5.24.0
ml SAMtools/1.6-GCCcore-6.3.0
ml kallisto/0.43.1-intel-2017A
ml R/3.4.2-intel-2017A-Python-2.7.12-default-mt
export R_LIBS=/scratch/user/dbrowne/Software/R_libs
#
python Cluster_Cutter_v1.py
```

The following Python script was developed and utilized to process the differentially expressed gene clusters and determine the gene functions associated with each cluster, as well as the set of non-differentially expressed genes:

```
#!/usr/bin/env python

import numpy as np
import pandas as pd

# Load and transform raw annotations into table of annotations per transcript

anno = pd.read_csv('./transcript.functions.txt', header=0, sep='\t')
tran = set(anno['#transcriptName'])
dtbs = set(anno['IdType'])
tble = pd.DataFrame(index=tran, columns=dtbs)

for i, r in anno.iterrows():
    t = r['#transcriptName']
    d = r['IdType']
    a = r['Id']
    if tble.ix[t, d] is np.NaN:
        tble.ix[t, d] = set([a])
    else:
        tble.ix[t, d].add(a)

# Load cluster data and parse into dictionary of transcripts per cluster and non-DE genes

clust = dict()
files = ["Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_10_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_11_log2_medianCentered_fpkm.matrix",
```

```
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_12_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_13_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_1_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_2_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_3_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_4_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_5_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_6_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_7_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_8_log2_medianCentered_fpkm.matrix",
        "Pt40_P1e-2C4/DGE_v1.P1e-
2C4.matrix.RData.clusters_fixed_P_40/subcluster_9_log2_medianCentered_fpkm.matrix"]

for f in files:
    n = 'DE_' + str(f.split('_')[-4])
    d = pd.read_csv(f, sep='\t', header=0, index_col=0)
    clust[n] = set(d.index)
de_genes = set([y for x in clust.values() for y in x])
non_de = tran - de_genes

# Create tables of function counts per cluster of genes

namer = {'EC': 'EC',
        'GO': 'GO',
        'KO': 'KEGGORTH',
        'PF': 'PFAM'}

def table_maker(db):
    tab = pd.DataFrame(columns=sorted(clust.keys(), key=lambda x: int(x.split('_')[1])))
    for k, v in sorted(clust.items(), key=lambda x: int(x[0].split('_')[1])):
        for g in v:
            if g in tble.index:
                if tble.ix[g, namer[db]] is not np.nan:
                    for f in tble.ix[g, namer[db]]:
                        try:
                            tab.loc[f, k] += 1
                        except KeyError:
                            tab.loc[f, k] = 1
    nd = dict()
    for i, r in tble.iterrows():
        if i in non_de and r[namer[db]] is not np.nan:
            for f in r[namer[db]]:
                try:
                    nd[f] += 1
                except KeyError:
                    nd[f] = 1
    tab['non_DE'] = [0] * len(tab)
    for f in tab.index:
        if f in nd:
            tab.loc[f, 'non_DE'] = nd[f]
    tab.fillna(0, inplace=True)
```

```
    tab.to_csv('./Cluster_' + db + '_Counts_Raw.csv')
    log = tab.apply(np.log1p)
    log.to_csv('./Cluster_' + db + '_Counts_Log.csv')

for db in namer:
    table_maker(db)
```

C.2.5 Transcription in Different Key Pathways

The following Python script was developed to access the KEGG API and extract pathway information, then parse the gene expression data for each pathway into separate tables for further downstream analysis:

```
#!/usr/bin/env python
#

import numpy as np
import pandas as pd
from Bio.KEGG import REST

# Load target KEGG pathway list and raw annotation data for B. braunii

target = open('./Pathway_List.txt', 'r').read().split('\n')[:-1]

anno = pd.read_csv('./transcript.functions.txt', header=0, sep='\t')
tran = set(anno['#transcriptName'])
dtbs = set(anno['IdType'])
tble = pd.DataFrame(index=tran, columns=dtbs)

for i, r in anno.iterrows():
    t = r['#transcriptName']
    d = r['IdType']
    a = r['Id']
    if tble.ix[t, d] is np.NaN:
        tble.ix[t, d] = set([a])
    else:
        tble.ix[t, d].add(a)

# Create dictionary that maps KEGG terms to transcripts

k_to_t = dict()

for i, r in tble.iterrows():
    if r['KEGGORTH'] is not np.nan:
        for k in r['KEGGORTH']:
            try:
                k_to_t[k].add(i)
            except KeyError:
                k_to_t[k] = set([i])

# Function to parse orthology terms into dictionary structure
```

```
def extract_orthology(pw):
    D = dict()
    for line in pw:
        if not line.startswith(' ') and 'ORTHOLOGY' in D:
            break
        elif 'ORTHOLOGY' in D:
            D['ORTHOLOGY'].append(line.split())
        elif line.startswith('ORTHOLOGY'):
            D['ORTHOLOGY'] = [line.split()[1:]]
    d = ['_'.join(x[1:]) if len(x[1:]) > 1 else x[1] for x in D['ORTHOLOGY']]
    e = [x[0] for x in D['ORTHOLOGY']]
    return zip(e, d)

# Load gene expression data and parse out pathway terms

expr = pd.read_csv('./Kv2A.isoform.counts.matrix', sep='\t', header=0, index_col=0)

for t in target:
    p = [x for x in REST.kegg_get(t)]
    k = extract_orthology(p)
    n = '_'.join(p[1].split()[1:]) if len(p[1].split()) > 2 else p[1].split()[1]
    with open('./Log/' + n + '_Log.csv', 'w') as o:
        o.write(',' + ','.join(expr.columns) + '\n')
        for e, d in k:
            if e in k_to_t:
                for g in k_to_t[e]:
                    if g in set(expr.index):
                        o.write(g + '_' + e + ',' + ','.join([str(np.log1p(x)) for x in
expr.loc[g]]) + '\n')
```

The following list of KEGG pathways was utilized in combination with the script described above:

```
ko00190
ko00195
ko00196
ko00710
ko00020
ko00030
ko00630
ko00061
ko00230
ko00240
ko00430
ko00780
ko00130
ko00860
ko00900
ko00902
ko00909
ko00906
ko03020
ko03022
ko03040
ko03010
ko00970
ko03013
```

ko03015
ko03050
ko03030
ko03440
ko04075
ko04712
ko04626
ko04113
ko04210
ko04016
ko02010
ko04120
ko04110
ko00940
ko00941

C.3 Materials and Methods for Analysis of Metabolite Profile

The code developed in this section was utilized to process the raw metabolomics data.

C.3.1 Targeted Analysis of Metabolite Profile

The total data in the targeted polar analysis (i.e. blanks, standards, experimental samples) were processed with the following Python script and then visualized with Morpheus:

```
#!/usr/bin/env python

import numpy as np
import pandas as pd


df4 = pd.read_csv('20171208_HILIC_POS_TD-
DB_Braunii_V2_2/sheets_POS_V2_2_AllSamples/peak_height.tab', sep='\t', header=0, skiprows=[1])
t = {x: x[40:] for x in df4.columns}
t['group'] = 'Metabolite'
df4.rename(columns=t, inplace=True)
df5 = df4.set_index('Metabolite')
df5.fillna(0, inplace=True)
df5.to_csv('./Targeted_Polar_Metabolites_v1.csv')
df6 = df5.apply(np.log1p)
df6.to_csv('./Targeted_Polar_Metabolites_v2.csv')
t2 = [x for x in df6.columns if 'Day' in x]
df7 = df6[t2]
df7.to_csv('./Targeted_Polar_Metabolites_v3.csv')
```

The following Python script was developed to process and filter the targeted polar metabolites to remove low-confidence metabolites identified in the experiment. This was achieved by comparing the experimental signals to the blank signals and removing metabolites where the blank signal was greater than 10% of the experimental signal.

```
#!/usr/bin/env python

import numpy as np
import pandas as pd

# Load raw data for targeted polar metabolites and write to file

df1 = pd.read_csv('20171208_HILIC_POS_TD-
DB_Braunii_V2_2/sheets_POS_V2_2_AllSamples/peak_height.tab', sep='\t', header=0, skiprows=[1])
t1 = {x: x[40:] for x in df1.columns}
t1['group'] = 'Metabolite'
df1.rename(columns=t1, inplace=True)
```

```
df2 = df1.set_index('Metabolite')
df2.fillna(0, inplace=True)

# Separate blank and experimental samples, filter data

t2 = [x for x in df2.columns if 'Day' in x]
t3 = [x for x in df2.columns if 'Blank' in x]

t4 = {'Day21-5AM':    '05_Day21_1',
      'Day21-5AM.1':  '05_Day21_2',
      'Day21-5AM.2':  '05_Day21_3',
      'Day22-5AM':    '05_Day22_1',
      'Day22-5AM.1':  '05_Day22_2',
      'Day22-5AM.2':  '05_Day22_3',
      'Day23-5AM':    '05_Day23_1',
      'Day23-5AM.1':  '05_Day23_2',
      'Day23-5AM.2':  '05_Day23_3',
      'Day21-5PM':    '17_Day21_1',
      'Day21-5PM.1':  '17_Day21_2',
      'Day21-5PM.2':  '17_Day21_3',
      'Day22-5PM':    '17_Day22_1',
      'Day22-5PM.1':  '17_Day22_2',
      'Day22-5PM.2':  '17_Day22_3',
      'Day23-5PM':    '17_Day23_1',
      'Day23-5PM.1':  '17_Day23_2',
      'Day23-5PM.2':  '17_Day23_3',
      'Day21-11AM':   '11_Day21_1',
      'Day21-11AM.1': '11_Day21_2',
      'Day21-11AM.2': '11_Day21_3',
      'Day22-11AM':   '11_Day22_1',
      'Day22-11AM.1': '11_Day22_2',
      'Day22-11AM.2': '11_Day22_3',
      'Day23-11AM':   '11_Day23_1',
      'Day23-11AM.1': '11_Day23_2',
      'Day23-11AM.2': '11_Day23_3',
      'Day21-11PM':   '23_Day21_1',
      'Day22-11PM':   '23_Day22_1',
      'Day22-11PM.1': '23_Day22_2',
      'Day22-11PM.2': '23_Day22_3',
      'Day23-11PM':   '23_Day23_1',
      'Day23-11PM.1': '23_Day23_2',
      'Day23-11PM.2': '23_Day23_3'}


df3 = pd.DataFrame(columns=sorted(t4.values()))

for i, r in df2.iterrows():
    me = float(np.mean([r[x] for x in t2]))
    mb = float(np.mean([r[x] for x in t3]))
    if mb < 0.1 * me and 0 not in set([r[x] for x in t2]):
        for x in t2:
            df3.loc[i, t4[x]] = r[x]

df3.to_csv('./Targeted_Polar_Metabolites_v4_Raw.csv')
df4 = df3.astype(np.float64).apply(np.log1p)
df4.to_csv('./Targeted_Polar_Metabolites_v4_Log.csv')
```

C.3.2 Untargeted Analysis of Metabolite Profile

The following Python script was developed to process the polar metabolomics data collected in positive ion mode:

```
#!/usr/bin/env python

import numpy as np
import pandas as pd

# Create dictionaries to simplify column naming

bla = {
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run1_171208155844.mzML filtered Peak height':
'MeOHBlank_Run1A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_25_B_Day23-
5AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run31.mzML filtered Peak height':
'MeOHBlank_Run31',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_20_B_Day22-
5PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run76.mzML filtered Peak height':
'MeOHBlank_Run76',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run6.mzML filtered Peak height':
'MeOHBlank_Run6',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_1_B_Day21-
5AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run13.mzML filtered Peak height':
'MeOHBlank_Run13',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_16_B_Day22-
11AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run19.mzML filtered Peak height':
'MeOHBlank_Run19',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_37_B_ExBlank_A_70to1050_MeOH_102030eV_MeOHBlank_Run10.mzML filtered Peak height':
'MeOHBlank_Run10',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_27_B_Day23-
5AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run118.mzML filtered Peak height':
'MeOHBlank_Run118',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_23_B_Day22-
11PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run67.mzML filtered Peak height':
'MeOHBlank_Run67',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_11_B_Day21-
11PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run49.mzML filtered Peak height':
'MeOHBlank_Run49',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_21_B_Day22-
5PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run115.mzML filtered Peak height':
'MeOHBlank_Run115',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_13_B_Day22-
5AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run37.mzML filtered Peak height':
'MeOHBlank_Run37B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_17_B_Day22-
11AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run55.mzML filtered Peak height':
'MeOHBlank_Run55',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_15_B_Day22-
5AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run112.mzML filtered Peak height':
'MeOHBlank_Run112',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run3.mzML filtered Peak height':
'MeOHBlank_Run3',
```

```
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Mid_MeOHBlank_____MeOHBlank_Run37.mzML filtered Peak height':
'MeOHBlank_Run37',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_39_B_ExBlank_C_70to1050_MeOH_102030eV_MeOHBlank_Run106.mzML filtered Peak height':
'MeOHBlank_Run106',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run2_171208185109.mzML filtered Peak height':
'MeOHBlank_Run2',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Mid_MeOHBlank_____MeOHBlank_Run91.mzML filtered Peak height':
'MeOHBlank_Run91',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_6_B_Day21-
11AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run100.mzML filtered Peak height':
'MeOHBlank_Run100',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_28_B_Day23-
11AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run34.mzML filtered Peak height':
'MeOHBlank_Run34',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_34_B_Day23-
11PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run40.mzML filtered Peak height':
'MeOHBlank_Run40',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_38_B_ExBlank_B_70to1050_MeOH_102030eV_MeOHBlank_Run52.mzML filtered Peak height':
'MeOHBlank_Run52',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_9_B_Day21-
5PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run97.mzML filtered Peak height':
'MeOHBlank_Run97',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_8_B_Day21-
5PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run64.mzML filtered Peak height':
'MeOHBlank_Run64',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run1.mzML filtered Peak height':
'MeOHBlank_Run1B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_3_B_Day21-
5AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run88.mzML filtered Peak height':
'MeOHBlank_Run88',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_29_B_Day23-
11AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run46.mzML filtered Peak height':
'MeOHBlank_Run46',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_36_B_Day23-
11PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run109.mzML filtered Peak height':
'MeOHBlank_Run109',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-
MSMS_37_B_ExBlank_A_70to1050_MeOH_102030eV_S1_Run8.mzML Peak height':
'ExBlank_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-
MSMS_38_B_ExBlank_B_70to1050_MeOH_102030eV_S1_Run50.mzML Peak height':
'ExBlank_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-
MSMS_39_B_ExBlank_C_70to1050_MeOH_102030eV_S1_Run102.mzML Peak height':
'ExBlank_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run7.mzML filtered Peak height':
'MeOHBlank_Run7',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_18_B_Day22-
11AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run85.mzML filtered Peak height':
'MeOHBlank_Run85',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_35_B_Day23-
11PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run61.mzML filtered Peak height':
'MeOHBlank_Run61',
```

```
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_26_B_Day23-
5AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run79.mzML filtered Peak height':
'MeOHBlank_Run79',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_7_B_Day21-
5PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run43.mzML filtered Peak height':
'MeOHBlank_Run43',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Mid_MeOHBlank_____MeOHBlank_Run64.mzML filtered Peak height':
'MeOHBlank_Run64B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_2_B_Day21-
5AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run58.mzML filtered Peak height':
'MeOHBlank_Run58',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_30_B_Day23-
11AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run94.mzML filtered Peak height':
'MeOHBlank_Run94',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_14_B_Day22-
5AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run82.mzML filtered Peak height':
'MeOHBlank_Run82',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_5_B_Day21-
11AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run73.mzML filtered Peak height':
'MeOHBlank_Run73',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_31_B_Day23-
5PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run22.mzML filtered Peak height':
'MeOHBlank_Run22',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_4_B_Day21-
11AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run25.mzML filtered Peak height':
'MeOHBlank_Run25',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_19_B_Day22-
5PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run16.mzML filtered Peak height':
'MeOHBlank_Run16',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_33_B_Day23-
5PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run91.mzML filtered Peak height':
'MeOHBlank_Run91B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_32_B_Day23-
5PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run70.mzML filtered Peak height':
'MeOHBlank_Run70',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_22_B_Day22-
11PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run28.mzML filtered Peak height':
'MeOHBlank_Run28',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_24_B_Day22-
11PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run105.mzML filtered Peak height':
'MeOHBlank_Run105',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Post_MeOHBlank_____MeOHBlank_Run121.mzML filtered Peak height':
'MeOHBlank_Run121',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run1_171208182052.mzML filtered Peak height':
'MOHBlank_Run1C'
}

exp = {
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_3_B_Day21-
5AM_C_70to1050_MeOH_102030eV_S1_Run86.mzML Peak height':       '05_Day21_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_20_B_Day22-
5PM_B_70to1050_MeOH_102030eV_S1_Run74.mzML Peak height':      '17_Day22_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_25_B_Day23-
5AM_A_70to1050_MeOH_102030eV_S1_Run29.mzML Peak height':      '05_Day23_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_30_B_Day23-
11AM_C_70to1050_MeOH_102030eV_S1_Run92.mzML Peak height':     '11_Day23_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_27_B_Day23-
5AM_C_70to1050_MeOH_102030eV_S1_Run116.mzML Peak height':     '05_Day23_C',
```

```
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_31_B_Day23-
5PM_A_70to1050_MeOH_102030eV_S1_Run20.mzML Peak height':      '17_Day23_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_29_B_Day23-
11AM_B_70to1050_MeOH_102030eV_S1_Run44.mzML Peak height':     '11_Day23_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_22_B_Day22-
11PM_A_70to1050_MeOH_102030eV_S1_Run26.mzML Peak height':     '23_Day22_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_8_B_Day21-
5PM_B_70to1050_MeOH_102030eV_S1_Run62.mzML Peak height':       '17_Day21_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_21_B_Day22-
5PM_C_70to1050_MeOH_102030eV_S1_Run113.mzML Peak height':     '17_Day22_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_18_B_Day22-
11AM_C_70to1050_MeOH_102030eV_S1_Run83.mzML Peak height':     '11_Day22_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_23_B_Day22-
11PM_B_70to1050_MeOH_102030eV_S1_Run65.mzML Peak height':     '23_Day22_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_33_B_Day23-
5PM_C_70to1050_MeOH_102030eV_S1_Run89.mzML Peak height':      '17_Day23_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_24_B_Day22-
11PM_C_70to1050_MeOH_102030eV_S1_Run101.mzML Peak height':    '23_Day22_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_16_B_Day22-
11AM_A_70to1050_MeOH_102030eV_S1_Run17.mzML Peak height':     '11_Day22_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_11_B_Day21-
11PM_B_70to1050_MeOH_102030eV_S1_Run47.mzML Peak height':     '23_Day21_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_32_B_Day23-
5PM_B_70to1050_MeOH_102030eV_S1_Run68.mzML Peak height':      '17_Day23_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_28_B_Day23-
11AM_A_70to1050_MeOH_102030eV_S1_Run32.mzML Peak height':     '11_Day23_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_15_B_Day22-
5AM_C_70to1050_MeOH_102030eV_S1_Run110.mzML Peak height':     '05_Day22_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_9_B_Day21-
5PM_C_70to1050_MeOH_102030eV_S1_Run95.mzML Peak height':       '17_Day21_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_5_B_Day21-
11AM_B_70to1050_MeOH_102030eV_S1_Run71.mzML Peak height':      '11_Day21_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_26_B_Day23-
5AM_B_70to1050_MeOH_102030eV_S1_Run77.mzML Peak height':      '05_Day23_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_34_B_Day23-
11PM_A_70to1050_MeOH_102030eV_S1_Run38.mzML Peak height':     '23_Day23_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_35_B_Day23-
11PM_B_70to1050_MeOH_102030eV_S1_Run59.mzML Peak height':     '23_Day23_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_13_B_Day22-
5AM_A_70to1050_MeOH_102030eV_S1_Run35.mzML Peak height':      '05_Day22_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_6_B_Day21-
11AM_C_70to1050_MeOH_102030eV_S1_Run98.mzML Peak height':      '11_Day21_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_17_B_Day22-
11AM_B_70to1050_MeOH_102030eV_S1_Run53.mzML Peak height':     '11_Day22_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_7_B_Day21-
5PM_A_70to1050_MeOH_102030eV_S1_Run41.mzML Peak height':       '17_Day21_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_14_B_Day22-
5AM_B_70to1050_MeOH_102030eV_S1_Run80.mzML Peak height':      '05_Day22_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_1_B_Day21-
5AM_A_70to1050_MeOH_102030eV_S1_Run11.mzML Peak height':       '05_Day21_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_4_B_Day21-
11AM_A_70to1050_MeOH_102030eV_S1_Run23.mzML Peak height':      '11_Day21_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_2_B_Day21-
5AM_B_70to1050_MeOH_102030eV_S1_Run56.mzML Peak height':       '05_Day21_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_19_B_Day22-
5PM_A_70to1050_MeOH_102030eV_S1_Run14.mzML Peak height':      '17_Day22_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_POS-MSMS_36_B_Day23-
11PM_C_70to1050_MeOH_102030eV_S1_Run107.mzML Peak height':    '23_Day23_C'
}
```

```
# Load dataframe of C18-NEG metabolomics data

raw = pd.read_csv('./20170919_KBL_C18_TD-
DB_Brauni/20171208_KBL_HILIC_Braunii_final_positive_formatted.csv', header=0)

# Load blank dataframe to store extracted data

pro = pd.DataFrame(columns=sorted(exp.values() + bla.values()))

# Transfer data into blank dataframe

for c in raw.columns:
    if c in exp:
        pro[exp[c]] = raw[c]
    elif c in bla:
        pro[bla[c]] = raw[c]

# Filter out data if mean blank signal > 0.1 * mean experimental signal

pro.fillna(0, inplace=True)
flt = pd.DataFrame(columns=sorted(exp.values()))

for i, r in pro.iterrows():
    mb = np.mean([r[x] for x in bla.values()])
    me = np.mean([r[x] for x in exp.values()])
    if mb < 0.1 * me and 0 not in set([r[x] for x in exp.values()]):
        for x in exp.values():
            flt.loc[i, x] = r[x]

# Write output to csv files

with open('./Processed_HILIC-POS_Data_v1_Raw.csv', 'w') as out_raw:
    flt.to_csv(out_raw)

with open('./Processed_HILIC-POS_Data_v1_Log.csv', 'w') as out_log:
    flt_log = flt.astype(np.float64).apply(np.log1p)
    flt_log.to_csv(out_log)
```

The following Python script was developed to process the polar metabolomics data collected in negative ion mode:

```
#!/usr/bin/env python

import numpy as np
import pandas as pd

# Column renaming dictionaries

bla = {
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_25_B_Day23-
5AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run31.mzML filtered Peak height':
'MeOHBlank_Run31',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_17_B_Day22-
11AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run55.mzML filtered Peak height':
'MeOHBlank_Run55',
```

'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_32_B_Day23-5PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run70.mzML filtered Peak height':
'MeOHBlank_Run70',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_24_B_Day22-11PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run105.mzML filtered Peak height':
'MeOHBlank_Run105',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_13_B_Day22-5AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run37.mzML filtered Peak height':
'MeOHBlank_Run37B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_3_B_Day21-5AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run88.mzML filtered Peak height':
'MeOHBlank_Run88',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_8_B_Day21-5PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run64.mzML filtered Peak height':
'MeOHBlank_Run64B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_0_Mid_MeOHBlank_____MeOHBlank_Run64.mzML filtered Peak height':
'MeOHBlank_Run64',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_34_B_Day23-11PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run40.mzML filtered Peak height':
'MeOHBlank_Run40',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_14_B_Day22-5AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run82.mzML filtered Peak height':
'MeOHBlank_Run82',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_29_B_Day23-11AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run46.mzML filtered Peak height':
'MeOHBlank_Run46',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_30_B_Day23-11AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run94.mzML filtered Peak height':
'MeOHBlank_Run94',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_22_B_Day22-11PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run28.mzML filtered Peak height':
'MeOHBlank_Run28',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_16_B_Day22-11AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run19.mzML filtered Peak height':
'MeOHBlank_Run19',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_21_B_Day22-5PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run115.mzML filtered Peak height':
'MeOHBlank_Run115',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_27_B_Day23-5AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run118.mzML filtered Peak height':
'MeOHBlank_Run118',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_1_B_Day21-5AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run13.mzML filtered Peak height':
'MeOHBlank_Run13',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_36_B_Day23-11PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run109.mzML filtered Peak height':
'MeOHBlank_Run109',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_4_B_Day21-11AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run25.mzML filtered Peak height':
'MeOHBlank_Run25',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_9_B_Day21-5PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run97.mzML filtered Peak height':
'MeOHBlank_Run97',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_33_B_Day23-5PM_C_70to1050_MeOH_102030eV_MeOHBlank_Run91.mzML filtered Peak height':
'MeOHBlank_Run91',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_0_Mid_MeOHBlank_____MeOHBlank_Run91.mzML filtered Peak height':
'MeOHBlank_Run91B',

'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_11_B_Day21-
11PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run49.mzML filtered Peak height':
'MeOHBlank_Run49',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_38_B_ExBlank_B_70to1050_MeOH_102030eV_MeOHBlank_Run52.mzML filtered Peak height':
'MeOHBlank_Run52',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_39_B_ExBlank_C_70to1050_MeOH_102030eV_MeOHBlank_Run106.mzML filtered Peak height':
'MeOHBlank_Run106',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run1_171208155844.mzML filtered Peak height':
'MeOHBlank_Run1B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Mid_MeOHBlank_____MeOHBlank_Run37.mzML filtered Peak height':
'MeOHBlank_Run37',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Post_MeOHBlank_____MeOHBlank_Run121.mzML filtered Peak height':
'MeOHBlank_Run121',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run2_171208185109.mzML filtered Peak height':
'MeOHBlank_Run2A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-
MSMS_38_B_ExBlank_B_70to1050_MeOH_102030eV_S1_Run51.mzML Peak height':
'ExBlank_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-
MSMS_39_B_ExBlank_C_70to1050_MeOH_102030eV_S1_Run104.mzML Peak height':
'ExBlank_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-
MSMS_37_B_ExBlank_A_70to1050_MeOH_102030eV_S1_Run9.mzML Peak height':
'ExBlank_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_20_B_Day22-
5PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run76.mzML filtered Peak height':
'MeOHBlank_Run76',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_23_B_Day22-
11PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run67.mzML filtered Peak height':
'MeOHBlank_Run67',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_7_B_Day21-
5PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run43.mzML filtered Peak height':
'MeOHBlank_Run43',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_26_B_Day23-
5AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run79.mzML filtered Peak height':
'MeOHBlank_Run79',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_35_B_Day23-
11PM_B_70to1050_MeOH_102030eV_MeOHBlank_Run61.mzML filtered Peak height':
'MeOHBlank_Run61',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_28_B_Day23-
11AM_A_70to1050_MeOH_102030eV_MeOHBlank_Run34.mzML filtered Peak height':
'MeOHBlank_Run34',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_19_B_Day22-
5PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run16.mzML filtered Peak height':
'MeOHBlank_Run16',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_15_B_Day22-
5AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run112.mzML filtered Peak height':
'MeOHBlank_Run112',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_2_B_Day21-
5AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run58.mzML filtered Peak height':
'MeOHBlank_Run58',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_5_B_Day21-
11AM_B_70to1050_MeOH_102030eV_MeOHBlank_Run73.mzML filtered Peak height':
'MeOHBlank_Run73',

```
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run7.mzML filtered Peak height':
'MeOHBlank_Run7',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_6_B_Day21-
11AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run100.mzML filtered Peak height':
'MeOHBlank_Run100',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_18_B_Day22-
11AM_C_70to1050_MeOH_102030eV_MeOHBlank_Run85.mzML filtered Peak height':
'MeOHBlank_Run85',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run6.mzML filtered Peak height':
'MeOHBlank_Run6',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-MS1_31_B_Day23-
5PM_A_70to1050_MeOH_102030eV_MeOHBlank_Run22.mzML filtered Peak height':
'MeOHBlank_Run22',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run3.mzML filtered Peak height':
'MeOHBlank_Run3',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_37_B_ExBlank_A_70to1050_MeOH_102030eV_MeOHBlank_Run10.mzML filtered Peak height':
'MeOHBlank_Run10',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run1_171208182052.mzML filtered Peak height':
'MeOHBlank_Run1A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_FPS-
MS1_0_Pre_MeOHBlank_____MeOHBlank_Run1.mzML filtered Peak height':
'MeOHBlank_Run1C'
}

exp = {
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_26_B_Day23-
5AM_B_70to1050_MeOH_102030eV_S1_Run78.mzML Peak height':    '05_Day23_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_14_B_Day22-
5AM_B_70to1050_MeOH_102030eV_S1_Run81.mzML Peak height':    '05_Day22_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_28_B_Day23-
11AM_A_70to1050_MeOH_102030eV_S1_Run33.mzML Peak height':   '11_Day23_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_3_B_Day21-
5AM_C_70to1050_MeOH_102030eV_S1_Run87.mzML Peak height':     '05_Day21_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_27_B_Day23-
5AM_C_70to1050_MeOH_102030eV_S1_Run117.mzML Peak height':   '05_Day23_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_35_B_Day23-
11PM_B_70to1050_MeOH_102030eV_S1_Run60.mzML Peak height':   '23_Day23_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_32_B_Day23-
5PM_B_70to1050_MeOH_102030eV_S1_Run69.mzML Peak height':    '17_Day23_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_25_B_Day23-
5AM_A_70to1050_MeOH_102030eV_S1_Run30.mzML Peak height':    '05_Day23_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_8_B_Day21-
5PM_B_70to1050_MeOH_102030eV_S1_Run63.mzML Peak height':     '17_Day21_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_20_B_Day22-
5PM_B_70to1050_MeOH_102030eV_S1_Run75.mzML Peak height':    '17_Day22_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_11_B_Day21-
11PM_B_70to1050_MeOH_102030eV_S1_Run48.mzML Peak height':   '23_Day21_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_21_B_Day22-
5PM_C_70to1050_MeOH_102030eV_S1_Run114.mzML Peak height':   '17_Day22_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_30_B_Day23-
11AM_C_70to1050_MeOH_102030eV_S1_Run93.mzML Peak height':   '11_Day23_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_13_B_Day22-
5AM_A_70to1050_MeOH_102030eV_S1_Run36.mzML Peak height':    '05_Day22_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_18_B_Day22-
11AM_C_70to1050_MeOH_102030eV_S1_Run84.mzML Peak height':   '11_Day22_C',
```

```
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_36_B_Day23-
11PM_C_70to1050_MeOH_102030eV_S1_Run108.mzML Peak height':  '23_Day23_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_24_B_Day22-
11PM_C_70to1050_MeOH_102030eV_S1_Run103.mzML Peak height':  '23_Day22_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_23_B_Day22-
11PM_B_70to1050_MeOH_102030eV_S1_Run66.mzML Peak height':   '23_Day22_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_31_B_Day23-
5PM_A_70to1050_MeOH_102030eV_S1_Run21.mzML Peak height':    '17_Day23_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_29_B_Day23-
11AM_B_70to1050_MeOH_102030eV_S1_Run45.mzML Peak height':   '11_Day23_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_6_B_Day21-
11AM_C_70to1050_MeOH_102030eV_S1_Run99.mzML Peak height':    '11_Day21_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_16_B_Day22-
11AM_A_70to1050_MeOH_102030eV_S1_Run18.mzML Peak height':   '11_Day22_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_34_B_Day23-
11PM_A_70to1050_MeOH_102030eV_S1_Run39.mzML Peak height':   '23_Day23_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_7_B_Day21-
5PM_A_70to1050_MeOH_102030eV_S1_Run42.mzML Peak height':     '17_Day21_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_9_B_Day21-
5PM_C_70to1050_MeOH_102030eV_S1_Run96.mzML Peak height':     '17_Day21_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_19_B_Day22-
5PM_A_70to1050_MeOH_102030eV_S1_Run15.mzML Peak height':    '17_Day22_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_22_B_Day22-
11PM_A_70to1050_MeOH_102030eV_S1_Run27.mzML Peak height':   '23_Day22_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_4_B_Day21-
11AM_A_70to1050_MeOH_102030eV_S1_Run24.mzML Peak height':    '11_Day21_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_33_B_Day23-
5PM_C_70to1050_MeOH_102030eV_S1_Run90.mzML Peak height':    '17_Day23_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_5_B_Day21-
11AM_B_70to1050_MeOH_102030eV_S1_Run72.mzML Peak height':    '11_Day21_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_1_B_Day21-
5AM_A_70to1050_MeOH_102030eV_S1_Run12.mzML Peak height':     '05_Day21_A',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_17_B_Day22-
11AM_B_70to1050_MeOH_102030eV_S1_Run54.mzML Peak height':   '11_Day22_B',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_15_B_Day22-
5AM_C_70to1050_MeOH_102030eV_S1_Run111.mzML Peak height':   '05_Day22_C',
'20171208_KBL_TD-DB_Bbraunii_Polar_Final_QE139-UV_HILIC_715831_NEG-MSMS_2_B_Day21-
5AM_B_70to1050_MeOH_102030eV_S1_Run57.mzML Peak height':     '05_Day21_B'
}


# Load dataframe of C18-NEG metabolomics data

raw = pd.read_csv('./20170919_KBL_C18_TD-
DB_Brauni/20171208_KBL_HILIC_Braunii_final_negative_formatted.csv', header=0)

# Load blank dataframe to store extracted data

pro = pd.DataFrame(columns=sorted(exp.values() + bla.values()))

# Transfer data into blank dataframe

for c in raw.columns:
    if c in exp:
        pro[exp[c]] = raw[c]
    elif c in bla:
        pro[bla[c]] = raw[c]

# Filter out data if mean blank signal > 0.1 * mean experimental signal

pro.fillna(0, inplace=True)
```

```
flt = pd.DataFrame(columns=sorted(exp.values()))

for i, r in pro.iterrows():
    mb = np.mean([r[x] for x in bla.values()])
    me = np.mean([r[x] for x in exp.values()])
    if mb < 0.1 * me and 0 not in set([r[x] for x in exp.values()]):
        for x in exp.values():
            flt.loc[i, x] = r[x]

# Write output to csv files

with open('./Processed_HILIC-NEG_Data_v1_Raw.csv', 'w') as out_raw:
    flt.to_csv(out_raw)

with open('./Processed_HILIC-NEG_Data_v1_Log.csv', 'w') as out_log:
    flt_log = flt.astype(np.float64).apply(np.log1p)
    flt_log.to_csv(out_log)
```

The following Python script was developed to process the nonpolar metabolomics data collected in positive ion mode:

```
#!/usr/bin/env python

import numpy as np
import pandas as pd

# Create dictionary to simplify column naming

columns = {
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_FPS-
MS1_0_Pre_____334Blank_Run7_170919144649.mzML filtered Peak height':        'Blank_Run',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_0_A1_ExBlank_B-
D__102040eV_132to1500_S1_Run8.mzML Peak height':          'A1_ExBlank',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_0_A2_ExBlank_B-
D__102040eV_132to1500_S1_Run69.mzML Peak height':         'A2_ExBlank',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_1_A_Day21-5AM_B-
D__102040eV_132to1500_S1_Run53.mzML Peak height':        '05_Day21_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_2_A_Day21-5AM_B-
D__102040eV_132to1500_S1_Run98.mzML Peak height':        '05_Day21_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_3_A_Day21-5AM_B-
D__205060eV_132to1500_S1_Run119.mzML Peak height':       '05_Day21_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_4_A_Day21-11AM_B-
D__102040eV_132to1500_S1_Run113.mzML Peak height':      '11_Day21_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_5_A_Day21-11AM_B-
D__102040eV_132to1500_S1_Run41.mzML Peak height':       '11_Day21_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_6_A_Day21-11AM_B-
D__205060eV_132to1500_S1_Run95.mzML Peak height':       '11_Day21_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_7_A_Day21-5PM_B-
D__102040eV_132to1500_S1_Run110.mzML Peak height':       '17_Day21_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_8_A_Day21-5PM_B-
D__102040eV_132to1500_S1_Run32.mzML Peak height':        '17_Day21_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_9_A_Day21-5PM_B-
D__205060eV_132to1500_S1_Run89.mzML Peak height':        '17_Day21_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_11_A_Day21-11PM_B-
D__102040eV_132to1500_S1_Run78.mzML Peak height':      '23_Day21_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_11_A_Day21-11PM_B-
D__205060eV_132to1500_S1_Run79.mzML Peak height':      '23_Day21_2',
```

```
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_13_A_Day22-5AM_B-
D__102040eV_132to1500_S1_Run104.mzML Peak height':     '05_Day22_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_14_A_Day22-5AM_B-
D__102040eV_132to1500_S1_Run14.mzML Peak height':      '05_Day22_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_15_A_Day22-5AM_B-
D__205060eV_132to1500_S1_Run62.mzML Peak height':      '05_Day22_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_16_A_Day22-11AM_B-
D__102040eV_132to1500_S1_Run47.mzML Peak height':     '11_Day22_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_17_A_Day22-11AM_B-
D__102040eV_132to1500_S1_Run26.mzML Peak height':     '11_Day22_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_18_A_Day22-11AM_B-
D__205060eV_132to1500_S1_Run20.mzML Peak height':     '11_Day22_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_19_A_Day22-5PM_B-
D__102040eV_132to1500_S1_Run101.mzML Peak height':     '17_Day22_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_20_A_Day22-5PM_B-
D__102040eV_132to1500_S1_Run50.mzML Peak height':      '17_Day22_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_21_A_Day22-5PM_B-
D__205060eV_132to1500_S1_Run23.mzML Peak height':      '17_Day22_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_22_A_Day22-11PM_B-
D__102040eV_132to1500_S1_Run107.mzML Peak height':    '23_Day22_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_23_A_Day22-11PM_B-
D__102040eV_132to1500_S1_Run92.mzML Peak height':     '23_Day22_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_24_A_Day22-11PM_B-
D__205060eV_132to1500_S1_Run29.mzML Peak height':     '23_Day22_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_25_A_Day23-5AM_B-
D__102040eV_132to1500_S1_Run44.mzML Peak height':      '05_Day23_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_26_A_Day23-5AM_B-
D__102040eV_132to1500_S1_Run86.mzML Peak height':      '05_Day23_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_27_A_Day23-5AM_B-
D__205060eV_132to1500_S1_Run11.mzML Peak height':      '05_Day23_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_28_A_Day23-11AM_B-
D__102040eV_132to1500_S1_Run38.mzML Peak height':     '11_Day23_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_29_A_Day23-11AM_B-
D__102040eV_132to1500_S1_Run17.mzML Peak height':     '11_Day23_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_30_A_Day23-11AM_B-
D__205060eV_132to1500_S1_Run116.mzML Peak height':    '11_Day23_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_31_A_Day23-5PM_B-
D__102040eV_132to1500_S1_Run75.mzML Peak height':      '17_Day23_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_32_A_Day23-5PM_B-
D__102040eV_132to1500_S1_Run83.mzML Peak height':      '17_Day23_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_33_A_Day23-5PM_B-
D__205060eV_132to1500_S1_Run35.mzML Peak height':      '17_Day23_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_34_A_Day23-11PM_B-
D__102040eV_132to1500_S1_Run59.mzML Peak height':     '23_Day23_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_35_A_Day23-11PM_B-
D__102040eV_132to1500_S1_Run56.mzML Peak height':     '23_Day23_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_POS-MSMS_36_A_Day23-11PM_B-
D__205060eV_132to1500_S1_Run72.mzML Peak height':     '23_Day23_3'
}

# Load dataframe of C18-NEG metabolomics data

raw = pd.read_csv('./20170919_KBL_C18_TD-DB_Brauni/20170919_KBL_C18_TD-
DB_BrauniiLipids_positive_formatted.csv', header=0)

# Load blank dataframe to store extracted data

pro = pd.DataFrame(columns=sorted(columns.values()))

# Transfer data into blank dataframe
```

```
for c in raw.columns:
    if c in columns:
        pro[columns[c]] = raw[c]

# Separate blank and experimental samples

bla = set([v for k, v in columns.items() if 'Blank' in k])
exp = set([v for k, v in columns.items() if 'Day' in k])

# Filter out data if mean blank signal > 0.1 * mean experimental signal

pro.fillna(0, inplace=True)
flt = pd.DataFrame(columns=sorted(exp))

for i, r in pro.iterrows():
    mb = float(np.mean([r[x] for x in bla]))
    me = np.mean([r[x] for x in exp])
    if mb < 0.1 * me and 0 not in set([r[x] for x in exp]):
        for x in exp:
            flt.loc[i, x] = r[x]

# Write output to csv files

with open('./Processed_C18-POS_Data_v2_Raw.csv', 'w') as out_raw:
    flt.to_csv(out_raw)

with open('./Processed_C18-POS_Data_v2_Log.csv', 'w') as out_log:
    flt_log = flt.astype(np.float64).apply(np.log1p)
    flt_log.to_csv(out_log)
```

The following Python script was developed to process the nonpolar metabolomics data collected in negative ion mode:

```
#!/usr/bin/env python

import numpy as np
import pandas as pd

# Column renaming dictionary

columns = {
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_FPS-
MS1_0_Pre_____334Blank_Run7_170919144649.mzML filtered Peak height': 'Blank_Run',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_0_A1_ExBlank_B-
D__102040eV_132to1500_S1_Run9.mzML Peak height': 'A1_ExBlank',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_0_A2_ExBlank_B-
D__102040eV_132to1500_S1_Run70.mzML Peak height': 'A2_ExBlank',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_1_A_Day21-5AM_B-
D__102040eV_132to1500_S1_Run54.mzML Peak height': '05_Day21_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_2_A_Day21-5AM_B-
D__102040eV_132to1500_S1_Run99.mzML Peak height': '05_Day21_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_3_A_Day21-5AM_B-
D__205060eV_132to1500_S1_Run120.mzML Peak height': '05_Day21_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_13_A_Day22-5AM_B-
D__102040eV_132to1500_S1_Run105.mzML Peak height': '05_Day22_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_14_A_Day22-5AM_B-
D__102040eV_132to1500_S1_Run15.mzML Peak height': '05_Day22_2',
```

'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_15_A_Day22-5AM_B-D__205060eV_132to1500_S1_Run63.mzML Peak height': '05_Day22_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_25_A_Day23-5AM_B-D__102040eV_132to1500_S1_Run45.mzML Peak height': '05_Day23_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_26_A_Day23-5AM_B-D__102040eV_132to1500_S1_Run87.mzML Peak height': '05_Day23_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_27_A_Day23-5AM_B-D__205060eV_132to1500_S1_Run12.mzML Peak height': '05_Day23_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_4_A_Day21-11AM_B-D__102040eV_132to1500_S1_Run114.mzML Peak height': '11_Day21_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_5_A_Day21-11AM_B-D__102040eV_132to1500_S1_Run42.mzML Peak height': '11_Day21_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_6_A_Day21-11AM_B-D__205060eV_132to1500_S1_Run96.mzML Peak height': '11_Day21_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_16_A_Day22-11AM_B-D__102040eV_132to1500_S1_Run48.mzML Peak height': '11_Day22_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_17_A_Day22-11AM_B-D__102040eV_132to1500_S1_Run27.mzML Peak height': '11_Day22_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_18_A_Day22-11AM_B-D__205060eV_132to1500_S1_Run21.mzML Peak height': '11_Day22_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_28_A_Day23-11AM_B-D__102040eV_132to1500_S1_Run39.mzML Peak height': '11_Day23_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_29_A_Day23-11AM_B-D__102040eV_132to1500_S1_Run18.mzML Peak height': '11_Day23_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_30_A_Day23-11AM_B-D__205060eV_132to1500_S1_Run117.mzML Peak height': '11_Day23_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_7_A_Day21-5PM_B-D__102040eV_132to1500_S1_Run111.mzML Peak height': '17_Day21_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_8_A_Day21-5PM_B-D__102040eV_132to1500_S1_Run33.mzML Peak height': '17_Day21_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_9_A_Day21-5PM_B-D__205060eV_132to1500_S1_Run90.mzML Peak height': '17_Day21_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_19_A_Day22-5PM_B-D__102040eV_132to1500_S1_Run102.mzML Peak height': '17_Day22_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_20_A_Day22-5PM_B-D__102040eV_132to1500_S1_Run51.mzML Peak height': '17_Day22_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_21_A_Day22-5PM_B-D__205060eV_132to1500_S1_Run24.mzML Peak height': '17_Day22_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_31_A_Day23-5PM_B-D__102040eV_132to1500_S1_Run76.mzML Peak height': '17_Day23_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_32_A_Day23-5PM_B-D__102040eV_132to1500_S1_Run84.mzML Peak height': '17_Day23_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_33_A_Day23-5PM_B-D__205060eV_132to1500_S1_Run36.mzML Peak height': '17_Day23_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_11_A_Day21-11PM_B-D__205060eV_132to1500_S1_Run81.mzML Peak height': '23_Day21_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_11_A_Day21-11PM_B-D__102040eV_132to1500_S1_Run80.mzML Peak height': '23_Day21_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_22_A_Day22-11PM_B-D__102040eV_132to1500_S1_Run108.mzML Peak height': '23_Day22_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_23_A_Day22-11PM_B-D__102040eV_132to1500_S1_Run93.mzML Peak height': '23_Day22_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_24_A_Day22-11PM_B-D__205060eV_132to1500_S1_Run30.mzML Peak height': '23_Day22_3',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_34_A_Day23-11PM_B-D__102040eV_132to1500_S1_Run60.mzML Peak height': '23_Day23_1',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_35_A_Day23-11PM_B-D__102040eV_132to1500_S1_Run57.mzML Peak height': '23_Day23_2',
'20170919_KBL_TD-DB_Bbraunii_Lipids_Final_QE139-UV_C18_102_NEG-MSMS_36_A_Day23-11PM_B-D__205060eV_132to1500_S1_Run73.mzML Peak height': '23_Day23_3',
}

```
# Load dataframe of C18-NEG metabolomics data

raw = pd.read_csv('./20170919_KBL_C18_TD-DB_Brauni/20170919_KBL_C18_TD-
DB_BrauniiLipids_negative_formatted.csv', header=0)

# Load blank dataframe to store extracted data

pro = pd.DataFrame(columns=sorted(columns.values()))

# Transfer data into blank dataframe

for c in raw.columns:
    if c in columns:
        pro[columns[c]] = raw[c]

# Separate blank and experimental samples

bla = set([v for k, v in columns.items() if 'Blank' in k])
exp = set([v for k, v in columns.items() if 'Day' in k])

# Filter out data if mean blank signal > 0.1 * mean experimental signal

pro.fillna(0, inplace=True)
flt = pd.DataFrame(columns=sorted(exp))

for i, r in pro.iterrows():
    mb = float(np.mean([r[x] for x in bla]))
    me = np.mean([r[x] for x in exp])
    if mb < 0.1 * me and 0 not in set([r[x] for x in exp]):
        for x in exp:
            flt.loc[i, x] = r[x]

# Write output to csv files

with open('./Processed_C18-NEG_Data_v3_Raw.csv', 'w') as out_raw:
    flt.to_csv(out_raw)

with open('./Processed_C18-NEG_Data_v3_Log.csv', 'w') as out_log:
    flt_log = flt.astype(np.float64).apply(np.log1p)
    flt_log.to_csv(out_log)
```